Low-Mach Number Flow and the Discontinuous Galerkin Method

Hennink, A.

DOI
10.4233/uuid:9b35b35a-0789-4883-a471-f8df0d7939ad

Publication date
2022

Document Version
Final published version

Citation (APA)
Hennink, A. (2022). *Low-Mach Number Flow and the Discontinuous Galerkin Method*. https://doi.org/10.4233/uuid:9b35b35a-0789-4883-a471-f8df0d7939ad

Low-Mach Number Flow and the Discontinuous Galerkin Method

Low-Mach Number Flow and the Discontinuous Galerkin Method

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, Prof. dr. ir. T. H. J. J. van der Hagen,
chair of the Board for Doctorates
to be defended publicly on
Tuesday 22 Februari 2022 at 10:00 o'clock

by

Aldo HENNINK

Master of Science in Applied Physics, Delft University of Technology, The Netherlands
born in Amsterdam, the Netherlands.

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof. dr. ir. J. L. Kloosterman	Delft University of Technology, promotor
Dr. ir. D. Lathouwers,	Delft University of Technology, promotor
Independent members:	
Prof. dr.-ing. E. Laurien,	University of Stuttgart, Germany
Prof. dr. ir. B. Koren,	Eindhoven University of Technology, the Netherlands
Prof. dr. S. Hickel,	Delft University of Technology
Dr. R. Pecnik,	Delft University of Technology
Prof. dr. P. Dorenbos,	Delft University of Technology, reserve member

Part of this work was funded by the sCO_2-HeRo project that has received funding from the European research and training program 2014–2018 (grant agreement ID 662116).

Part of this work was supported by the ENEN+ project that has received funding from the Euratom research and training Work Programme 2016–2017 — 1 #755576.

Keywords: Low-Mach Number Flow, Pressure Correction, Discontinuous Galerkin

Contents

Nomenclature vii

Summary ix

Samenvatting xi

1 Introduction 1
- 1.1 Mathematical Problem Setting . . . 1
- 1.2 Overview of the Numerical Method . . . 3
  - 1.2.1 Why Use a Discontinuous Galerkin Method? . . . 4
- 1.3 Thesis Overview . . . 6

References . . . 6

2 Spatial Discretization with a Discontinuous Galerkin Method 9
- 2.1 Introduction . . . 9
  - 2.1.1 Overview of DG discretizations . . . 9
  - 2.1.2 Construction of the Solution Space . . . 10
- 2.2 Discrete Continuity Equation . . . 12
- 2.3 Discrete Momentum Equation . . . 14
  - 2.3.1 Discretization of the Viscous Stress . . . 14
- 2.4 Discrete Enthalpy Equation . . . 16
- 2.5 Discretization of the Convection . . . 16
  - 2.5.1 Choice of the Numerical Flux . . . 17
  - 2.5.2 Solution Spaces for the Enthalpy and the Pressure . . . 19
  - 2.5.3 Proper Treatment of Dirichlet Boundary Conditions . . . 20
- 2.6 Implementation . . . 22
- 2.7 Test Case: A Heated Backward-facing Step . . . 23
- 2.8 Discussion and Conclusion . . . 26

References . . . 27

3 Pressure Correction 33
- 3.1 Introduction . . . 33
  - 3.1.1 Fully Discrete Linear System . . . 33
- 3.2 Pressure Correction Method . . . 35
- 3.3 Verification with Manufactured Solutions . . . 38
  - 3.3.1 Taylor-Green Vortex . . . 38
  - 3.3.2 Variable-property Manufactured Solution . . . 39
- 3.4 Validation with Flow Past a Circular Obstacle . . . 42
- 3.5 Discussion . . . 46
  - 3.5.1 Pressure Correction with Equal-order Discretizations . . . 48

References . . . 49

4 Handling the Enthalpy Equation for Low-Mach Number Flow 53
4.1 Introduction . . . 53
4.1.1 In Between Compressible and Incompressible . . . 53
4.1.2 Which Enthalpy Equation Should be Solved? . . . 54
4.2 The Temporal Density Gradient . . . 57
4.3 Linearizing the Temporal Derivative of the Enthalpy . . . 58
4.3.1 Error Estimates and Stability . . . 59
4.3.2 Proper Scaling of the Enthalpy Equation . . . 61
4.3.3 Special Case of an Ideal Gas . . . 62
4.4 Test Case for the Space-independent Enthalpy Equation . . . 63
4.5 Test Cases with Low-Mach Number Flow . . . 65
4.5.1 Variable-density Manufactured Solution . . . 65
4.5.2 Validation with Flow Past a Heated Circular Obstacle . . . 72
4.6 Discussion and Conclusion . . . 73

Appendices 76
A Derivations of the Results in Section 4.3.1 . . . 76
A.1 Derivations for Method #1 . . . 76
A.2 Derivations for Method #2 . . . 78
References . . . 80

5 Channel Flow and Large Eddy Simulation 83
5.1 Introduction and Governing Equations . . . 83
5.2 Sub-filter Scale models . . . 86
5.3 Numerical Simulation . . . 89
5.3.1 Discretization . . . 89
5.3.2 Including a Variable Density . . . 91
5.4 Test Case: Infinite Plane Channel Flow . . . 92
5.4.1 Dimensionless Analysis . . . 93
5.4.2 Initial Condition . . . 94
5.4.3 Domain Size and Mesh . . . 94
5.4.4 Results . . . 96
5.5 Discussion . . . 97
References . . . 101

6 Conclusion 107

Acknowledgements 109

Curriculum Vitæ 111

List of Publications 113

Nomenclature

General

∇_i	gradient in the i^{th} Cartesian coordinate (e.g., $\nabla_2 \coloneqq \partial/\partial y$)
ρ_T, h_ρ, H_h, …	partial derivatives with respect to fluid properties (e.g., $\rho_T \coloneqq \partial\rho/\partial T$)
$\mathbf{u}$, $\mathbf{F}$, …	vector-values quantities (e.g., velocity, force,)
$\underline{\boldsymbol{m}}$, $\underline{\boldsymbol{h}}$, $\underline{\boldsymbol{p}}$, …	vectors in the discrete global linear system
$\boldsymbol{M}$, $\boldsymbol{D}$, $\boldsymbol{C}$, …	matrices in the discrete global linear system

List of symbols

Independent variables and domain:

$\mathbf{r} = [x, y, z]$	Cartesian spatial coordinate
t	time
Ω	spatial domain
$\mathbf{n}$	outward normal vector
$\partial\Omega$, ∂T, …	boundaries of Ω, T, …
$\partial\Omega^{\mathrm{D}}$, $\partial\Omega^{\mathrm{N}}$	parts of $\partial\Omega$ with a Dirichlet (D) or Neumann (N) boundary condition

Related to the spatial discretization:

$\mathcal{T}$	the set of elements (i.e., the computational mesh)
$\mathcal{F}$	the set of all faces in the mesh
$\mathcal{F}^{\mathrm{i}}$, $\mathcal{F}^{\mathrm{D}}$, $\mathcal{F}^{\mathrm{N}}$	sets of all internal (i), Dirichlet (D), or Neumann (N) faces
$\mathcal{P}_X$	polynomial order of a numerical solution space of quantity X

Variables in the transport equations:

p^{th}	thermodynamic pressure
p	'incompressible' (or 'hydrodynamic', or 'mechanical') pressure
$\mathbf{m}$	mass flux
$\mathbf{u} \coloneqq (1/\rho)\mathbf{m}$	velocity
τ	viscous stress tensor
$\mathbf{F}$	volumetric external force
Q	volumetric heat source
ϕ^{D}	Dirichlet value of a quantity ϕ at a boundary (i.e., the inhomogeneous part of a Dirichlet boundary condition)
$\mathbf{f}^{\mathrm{N}}$	stress at the outflow boundary (i.e., the inhomogeneous part of a Neumann boundary condition for the momentum)

q^{N}	heat flux out of the domain at the boundary (i.e., the inhomogeneous part of a Neumann boundary condition for the enthalpy)

Fluid properties:

T	temperature
ρ	density
h	specific enthalpy
$H = \rho h$	volumetric enthalpy
h_0	enthalpy offset
$c_p \coloneqq h_T$	specific heat capacity at constant thermodynamic pressure
k	thermal conductivity
$\alpha \coloneqq k/(\rho c_p)$	thermal diffusivity
μ	dynamic viscosity
$\nu \coloneqq \mu/\rho$	kinematic viscosity

Dimensionless numbers:

$\mathrm{Pr} \coloneqq \nu/\alpha$	Prandtl number
Re	Reynolds number
Nu	Nusselt number
f_{D}	Darcy friction factor
St	Strouhal number

Summary

This thesis describes a numerical method for computational fluid dynamics. Special attention is paid to low-Mach number flows.

The spatial discretization is a discontinuous Galerkin method, based on modal basis functions. The convection is discretized with the local Lax-Friedrichs flux. The diffusion in the enthalpy equation is discretized with the symmetric interior penalty method, which is generalized in a straightforward manner for the viscous stress in the momentum equation. The numerical method does not deviate fundamentally from previous literature.

The temporal derivatives in the enthalpy and momentum equations are discretized with a second-order backward finite difference method. An algorithmic pressure correction scheme is used decouple the momentum and the continuity equations, giving rise to explicit artificial boundary conditions. If the pressure and the momentum are discretized with an equal-order polynomial space, then the pressure equation is stabilized with an extra penalty term to suppress the discontinuities in the solution, as explained in chapter 2.

Using a time-splitting method is far more difficult when the flow is compressible, due the variable density. Low-Mach number flows also do not lend themselves well to solving the coupled transport equations, because the density is a function of the enthalpy, not the pressure. This differs from high-Mach number flows, where one can solve a transport equation for the density. Chapter 4 describes in great detail how a non-constant density can be incorporated into a time-splitting scheme for low-Mach number flows.

Chapter 4 also discusses the best form of the enthalpy transport equation to solve (primitive or conservative), and for which variable (primitive or conserved). This question arises in low-Mach number flows, because the density is a function of the temperature. Here the conservative transport equation is solved for the specific enthalpy.

The main difficulty with this approach is that the temporal enthalpy derivative is nonlinear due to the variable density. This can be addressed with an easily implemented adjustment of the finite difference scheme ('method #2' in sections 4.3–4.4). The resulting discretization displays second-order temporal accuracy (as measured in the spatial L^2 norm) for the enthalpy and the mass flux, without having to iterate within a time step.

Furthermore, the enthalpy transport equation needs to be stabilized with a simple change of variables, in which the specific enthalpy is 'offset' by a constant. Though it may be counter-intuitive, the enthalpy offset is critical to the stability and the accuracy of the temporal discretization. This would also be true if one were to solve for the volumetric enthalpy, because the enthalpy offset determines whether there is a one-to-one mapping between the volumetric enthalpy and the density.

The spatial and temporal discretizations and their implementations are extensively verified and validated with the test cases at the end of the chapters. In particular, sections 3.3.1, 3.3.2, and 4.5.1 feature exhaustive tests with manufactured solutions with nontrivial fluid properties. Sections 2.7, 3.4, and 4.5.2 contain validations for laminar flows. Chapter 5 also shows simulations of turbulent flows.

Samenvatting

Deze scriptie beschrijft een numerieke methode voor vloeistofdynamica. Er gaat extra aandacht uit naar stromingen met een laag Mach-getal.

De spatiële discretisatie is een discontinue Galerkin-methode, gebaseerd op modale basisfuncties. Voor de discretisatie van de convectie wordt gebruikt gemaakt van de Lax-Friedrichs-flux. Voor de discretisatie van de diffusieve term in de enthalpievergelijking wordt gebruikt gemaakt van de 'symmetric interior pentaly'-methode, die in een algemenere vorm ook toepasbaar is op de viskeuze term in the impulsvergelijking. De numerieke method wijkt niet fundamenteel af van voorgaande literatuur.

The tijdsafgeleide in de enthalpie- en impulsvergelijkingen worden discreet gemaakt met een impliciete eindigverschilmethode van tweede orde. Een algoritmisch drukcorrectieschema ontkoppelt de drukvergelijking en de impulsvergelijking, wat resulteert in expliciete artificiële randvoorwaarden. Als voor de druk en de impuls een polynome oplossingsruimte van gelijke order wordt gebruikt, dan moet de drukvergelijking gestabiliseerd worden met een extra 'penalty'-term om de discontinuïteiten in de oplossing in toom te houden, zoals wordt uitgelegd in hoofdstuk 2.

In een compressibele stroming wordt de tijdsplitsingsmethode aanzienlijk bemoeilijkt door de variabele dichtheid. Voor stromingen met een laag Mach-getal is het ook niet makkelijk om de gekoppelde transportvergelijkingen op te lossen, omdat de dichtheid een functie is van de enthalpie, niet van de druk. Hoofdsuk 4 detailleert hoe een niet-constante dichtheid opgenomen kan worden in een tijdsplitsingsschema voor stromingen met een laag Mach-getal.

Hoofdstuk 5 behandelt ook in welke vorm de enthalpievergelijking het best kan worden opgelost (primitief of conservatief) en voor welke variabele (de specifieke of de volumetrische enthalpie). Deze vraagstukken spelen een rol bij stromingen met een laag Mach-getal, omdat de dichtheid een functie is van de temperatuur. In dit werk wordt de conservatieve enthalpievergelijking opgelost voor de specifieke enthalpie.

De grootste uitdaging bij deze aanpak is de niet-lineaire tijdsafgeleide van de enthalpie, als gevolg van de variabele dichtheid. Dit probleem kan worden geadresseerd met een eenvoudig te implementeren aanpassing van de eindigverschilmethode ('methode #2' in sections 4.3–4.4). Dit resulteert in tweede-order nauwkeurigheid in de tijd (gemeten in de spatiële L^2-norm) voor de enthalpie en de massaflux, zonder te hoeven itereren binnen een tijdstap.

Daarnaast moet de enthalpievergelijking gestabiliseerd worden met een eenvoudige substitutie van variabelen, waarbij een constante bij de specifieke enthalpie wordt opgeteld. Deze 'enthalpieverspringing' ('enthalpy offset') is essentieel voor de stabiliteit en de nauwkeurigheid van de discretisatie. Dat zou ook het geval

geweest zijn als we de volumetrische enthalpie zouden hebben gekozen als onbekende variabele, omdat de enthalpieverspringing bepaalt of er één-op-één-verband bestaat tussen de volumetrische enthalpie en de dichtheid.

De discretisaties in ruimte en tijd worden uitvoerig geverifieerd en gevalideerd met de tests aan het einde van de hoofdstukken. Hierbij zijn vooral de tests met artificiële oplossingen en niet-triviale vloeistofeigenschappen noemenswaardig, zie secties 3.3.1, 3.3.2 and 4.5.1. Secties 2.7, 3.4 and 4.5.2 behandelen validaties met laminaire stromingen. Hoofdstuk 5 toont ook simulaties met turbulente stromingen.

1

Introduction

This thesis is on computational fluid dynamics for incompressible flows and flows in the low-Mach number limit. Various numerical issues are explored, related to the spatial discretization, the time-splitting method, and turbulent flows. Special attention is paid to how the discretizations can be adjusted when the density depends on the temperature.

These are fairly unrelated topics, and it would be infeasible to cover all the basics in a single chapter. The following chapters therefore contain their own introductions, including extensive literature reviews. Here we merely state the mathematical problem (i.e., the governing equations with boundary conditions).

It is assumed throughout the thesis that the reader is familiar with common concepts in computational fluid dynamics.

This text is also not meant as an introduction to the discontinuous Galerkin (DG) method, though section 1.2 discusses some reasons for its recent gain in popularity, and can be read without prior knowledge. Chapter 2 ('Spatial Discretization with a Discontinuous Galerkin Method') is technically self-contained, but focuses on what differs from other literature, while skipping over some important basics. The reader can instead consult one of several monographs, of which the one by Hesthaven and Warburton [1] likely provides the gentlest introduction.

1.1. Mathematical Problem Setting

The low-mach number limit of the compressible flow equations can be obtained in a conceptually straightforward manner by expanding each variable ϕ into a Maclaurin series in the Mach number Ma, that is, $\phi = \phi^{(0)} + \phi^{(1)}\mathrm{Ma} + \phi^{(2)}\mathrm{Ma}^2 + \ldots$. Since the equations must hold for all Ma, this leads to a series of equations, corresponding to the coefficients of Ma^0, Ma^1, Ma^2, ... The terms of order $\mathcal{O}\left(\mathrm{Ma}^3\right)$ are neglected. A detailed derivation can be found in several places, including the seminal paper on this approach by Paolucci [2].

1

In the resulting transport equations, the pressure is split into two parts: the thermodynamic pressure p^{th} (of order Ma^0), which is used to evaluate the equation of state, and a pressure p (of order Ma^2) in the momentum equation that acts as a Lagrange multiplier to enforce continuity, just like in the incompressible flow equations [2]. We simply refer to p as the pressure; it has also been called the 'hydrodynamic pressure'[1] (e.g, [3, 4]), the 'mechanical pressure' (e.g., [5]), and the 'incompressible pressure' (e.g., [6]).

Assuming a constant thermodynamic pressure, the transport equations in the low-Mach number limit are

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{m} = 0 \,, \tag{1.1a}$$

$$\frac{\partial \mathbf{m}}{\partial t} + \nabla \cdot (\mathbf{u}\ \mathbf{m}) = \nabla \cdot \tau - \nabla p + \mathbf{F} \,, \tag{1.1b}$$

$$\frac{\partial \rho h}{\partial t} + \nabla \cdot (\mathbf{m} h) = -\nabla \cdot \mathbf{q} + Q \tag{1.1c}$$

on a domain Ω. Here t is the time, ρ is the density, $\mathbf{u}$ is the velocity, $\mathbf{m} \coloneqq \rho \mathbf{u}$ is the mass flux, h is the specific enthalpy, and $\mathbf{F}$ and Q are known external sources. The pressure p has no effect on any of the fluid properties.

The general form of the low-Mach number equations has a non-constant thermodynamic pressure $p^{\mathrm{th}} = p^{\mathrm{th}}(t)$ and a density that is a function of both p^{th} and h. The enthalpy equation then gets an extra term $\mathrm{d}p^{\mathrm{th}}/\mathrm{d}t$, which can be estimated by integrating the enthalpy equation. Since all variable-density flows in this work have an outlet, where p^{th} is fixed, we simply have $\mathrm{d}p^{\mathrm{th}}/\mathrm{d}t = 0$.

Assuming a Newtonian fluid, the stress tensor is

$$\tau_{ij} = \mu \left(\nabla_i u_j + \nabla_j u_i - \frac{2}{3} (\nabla \cdot \mathbf{u}) \, \delta_{ij} \right) \,. \tag{1.2}$$

The heat flux is

$$\mathbf{q} = -k \nabla T = -\frac{k}{c_p} \nabla h \,, \tag{1.3}$$

where T is the temperature. k is the thermal conductivity, and $c_p \coloneqq \partial h / \partial T$ is the specific heat capacity. The last equality is technically an approximation because it neglects the dependence of the temperature on the pressure, but this is highly accurate for low-Mach number flows, even for strongly variable fluid properties in supercritical fluids [3]. The transport properties (μ, k) and the specific heat capacity (c_p) are a function of T, but do not depend significantly on the pressure.

We consider two types of boundaries:

- Dirichlet boundaries, denoted by $\partial\Omega^{\mathrm{D}}$, on which the mass flux and the temperature (and, therefore, the enthalpy and the fluid properties) are given, that is, $\mathbf{m} = \mathbf{m}^{\mathrm{D}}$ and $T = T^{\mathrm{D}}$;

[1] This is perhaps slightly confusing because of the double meaning of the 'dynamic pressure' in the Bernoulli equations.

- outflow boundaries, denoted by $\partial\Omega^{\mathrm{N}}$, where

$$(\tau - pI)\cdot\mathbf{n} = \mathbf{f}^{\mathrm{N}} \quad \text{and} \quad k\nabla T\cdot\mathbf{n} = q^{\mathrm{N}} \tag{1.4}$$

  are prescribed.

Periodic boundary conditions do not require special attention, because corresponding mesh elements at opposite periodic boundaries can simply be treated as internal elements. Of course the mass flux and the temperature must also be equipped with appropriate initial conditions, but these will not play a significant role.

1.2. Overview of the Numerical Method

The goal of a numerical method is to find an approximate solution to the system 1.1a–1.1c. Since computers work efficiently with blocks of data, it is natural to represent the state of a fluid by a list of numbers, which form a vector $\underline{\boldsymbol{\phi}} \in \mathbb{R}^N$. This is known as a discretization, because it transforms the continuous problem into a problem with a countable number (N) of degrees of freedom. In this thesis the terms 'numerical method' and 'discretization' are used interchangeably.

The type of discretization is determined by the way in which $\underline{\boldsymbol{\phi}}$ represents an unknown continuous state. For the spatial discretization we use a finite element method, which is based on N predefined basis functions $\{\xi_i\}_{i=1}^N$. A generic exact quantity $\tilde{\phi}$ is approximated by the linear combination $\tilde{\phi} \approx \phi = \underline{\boldsymbol{\phi}}_k \xi_k$. In other words, the coefficients of the solution vector $\underline{\boldsymbol{\phi}}$ are the weights of the basis functions.

An effective way to use the computing power is to solve a linear system

$$\boldsymbol{A}\underline{\boldsymbol{\phi}} = \underline{\boldsymbol{b}} \tag{1.5}$$

for $\underline{\boldsymbol{\phi}}$, which is therefore known as the solution vector. In practice such a large linear system can only be solved approximately, and there is an enormous body of research on how this can be done. This typically has a large impact on the overall efficacy of the numerical method, but that is not the topic of this thesis. Section 2.6 outlines how we solve the linear equations, and we make occasional reference to the linear solvers when they are relevant to the rest of the discretization (such as in section 3.5.1), but the reader can assume that the above equation is solved with a negligible error.

That leaves the question of how to construct the matrix $\boldsymbol{A}$ and right-hand side $\underline{\boldsymbol{b}}$. In a finite element method this is done with discrete bilinear and linear operators $a(\cdot,\cdot)$ and $b(\cdot)$ that generate the entries in $\boldsymbol{A}$ and $\underline{\boldsymbol{b}}$ with the basis functions, that is,

$$\boldsymbol{A}_{ij} = a(\xi_i, \xi_j) \quad \text{and} \quad \underline{\boldsymbol{b}}_i = b(\xi_i)\ . \tag{1.6}$$

The linear system is equivalent to the problem

$$\begin{aligned} &\textit{Find } \underline{\boldsymbol{\phi}} \in \mathbb{R}^N\textit{, such that, for all } \underline{\boldsymbol{v}} \in \mathbb{R}^N\textit{,} \\ &\underline{\boldsymbol{v}}^{\mathsf{T}}\boldsymbol{A}\underline{\boldsymbol{\phi}} = \underline{\boldsymbol{v}}^{\mathsf{T}}\underline{\boldsymbol{b}}\ . \end{aligned} \tag{1.7}$$

The test vector $\underline{\boldsymbol{v}}$ has a corresponding test function $v = \underline{\boldsymbol{v}}_k \, \xi_k$. Using the linearity of a and b, we have $\underline{\boldsymbol{v}}^\intercal \underline{\boldsymbol{A}} \underline{\boldsymbol{\phi}} = \underline{\boldsymbol{v}}_k a\left(\xi_k, \xi_q\right) \underline{\boldsymbol{\phi}}_q = a\left(\underline{\boldsymbol{v}}_k \xi_k \, , \, \underline{\boldsymbol{\phi}}_q \, \xi_q\right) = a\left(v, \phi\right)$ and $\underline{\boldsymbol{v}}^\intercal \underline{\boldsymbol{b}} = \underline{\boldsymbol{v}}_k b\left(\xi_k\right) = b\left(\underline{\boldsymbol{v}}_k \xi_k\right) = b\left(v\right)$, so the linear system can also be written as

$$\begin{aligned} &\textit{Find } \phi \in V_\phi\textit{, such that, for all } v \in V_\phi\textit{,} \\ &a(\phi, v) = b(v) \ , \end{aligned} \tag{1.8}$$

where $V_\phi = \operatorname{span}\left(\{\xi_i\}_{i=1}^N\right)$ is the numerical solution space (i.e., the span of all basis functions). Instead of deriving the linear system directly, we work with statements like Eq. 1.8, which is known as the discrete weak form.

The precise type of finite element method we use is a discontinuous Galerkin (DG) method, characterized by the fact that the basis functions are not continuous, and neither is the numerical approximation. The term 'Galerkin' refers to the fact that the test function in Eq. 1.8 lies in the same solution space as the numerical solution. As a result of the discontinuities, the DG method shares many characteristics with the finite volume method, especially when applied to hyperbolic problems, as will become clear in Chapter 2.

In practice we have to decouple the spatial and temporal discretizations to make the computations feasible. For the temporal discretization we use a finite difference method, resulting in solution vectors $\left\{\underline{\boldsymbol{\phi}}^n, \underline{\boldsymbol{\phi}}^{n+1}, \ldots\right\}$ to approximate the solution at discrete times $\left\{t^n, t^{n+1}, \ldots\right\}$. We also split the full coupled state of the fluid into three separate solution vectors for the mass flux, pressure, and enthalpy ($\underline{\boldsymbol{m}}$, $\underline{\boldsymbol{p}}$, and $\underline{\boldsymbol{h}}$), as is common for incompressible flows.

1.2.1. Why Use a Discontinuous Galerkin Method?

The discontinuous Galerkin method initially received little attention outside of the field of particle transport, where it was introduced in 1973 [7]. This has changed over the last one or two decades, and the DG method is now a very active area of research with applications in many fields, including computational fluid dynamics. Each author seems to prefer the DG method for his or her own reasons; the literature does not agree on the most important advantage over other discretizations. We list a few possible advantages here.

Early work by Collis [8] focused on the weak imposition of Dirichlet boundary conditions in an attempt to explain their remarkably accurate results with few degrees of freedom. Ern and Guermond [9] provide a more theoretical analysis of weak Dirichlet boundary conditions for Friedrich's system. Weak boundary conditions have long been known to be superior to strongly imposed boundary conditions, where the numerical solution satisfies the boundary condition at every point on the boundary, as had been common in classical finite element methods. A weak boundary condition can be shown to act as an implicit filter in badly resolved flows [10].

Weak Dirichlet boundary conditions with a user-defined penalty parameter also provided the inspiration for the first penalty methods to discretize elliptic problems

with a discontinuous solution space. The unified analysis of DG methods for elliptic problems by Arnold *et al.* [11] was foundational for extending the DG method to viscous flow. The required penalty parameter has been estimated by many authors, including Shahbazi [12]. They later also established the DG method for convection-diffusion problems for incompressible flow with a time-splitting method [13].

An interesting property of the DG solution space is that it supports a low-pass filter in a natural way, by separating the high-order and low-order polynomials in the solution space, giving it great potential for a variational multi-scale method. Collis [14] has discussed this as early as 2002. See Hughes *et al.* [15] for a comparison of continuous and discontinuous solution spaces in this context. Variational multiscale with DG is still an ongoing area of research (e.g., [16, 17]).

There are other applications for the natural support for scale separation. For example, Atkins and Helenbrook [18] and Helenbrook and Atkins [19] have introduced polynomial-based multigrid methods to DG discretizations in 2005. Separating the polynomials can also be used for the coarse-grid projection in a dynamic Large Eddy simulation [20].

Most authors recognize the arbitrarily high order of accuracy on unstructured meshes as a major benefit of DG discretizations. Several papers have applied the DG method to nontrivial geometries, such as airfoils. This geometric flexibility could be valuable for some industrial applications (as discussed in, e.g., [21]). It also provides support for adaptivity (e.g., [22]). Furthermore, the unstructured nature of DG methods has been used for arbitrary space-time domains and moving boundary problems (e.g., [23–26]).

The most important numerical characteristic of any discretization is perhaps how well it handles unresolved flows. Turbulent flow simulations are rarely fully resolved, and that is particularly true for large eddy simulations (LES), where it is standard practice to rely on a moderate amount of numerical dissipation. This is likely to increase in importance, as more practical flows are starting to fall within the range of applications of LES.

There are indications that the DG method handles unresolved flows well. One reason is that the weak form in a DG method can be written as a local conservation law for each element in the mesh, much like in finite volume methods. This offers stability for badly resolved flows, despite the high order of accuracy. Another reason is the frequency spectrum of the numerical dissipation, which many authors have studied recently (e.g., [27]). High-order DG discretizations acts as a low-pass filter. Moura *et al.* [28] have argued that this makes them particularly suitable for implicit large eddy simulations.

More recently, there has been much attention to how the computational structure of DG discretizations can be implemented efficiently on modern hardware architectures. The DG method results in a block matrix, wherein the blocks are coupled with a compact stencil. Chapelier *et al.* [17] have claimed that this is beneficial for MPI parallelization. The block structure also lends itself to acceleration with GPUs (e.g., [29]). As mentioned by Kronbichler *et al.* [30], 'the most common obstacle to high-order methods is a lack of competitive implementations in generic numerical software packages, which has limited their application mostly to specialized codes

targeting'. This makes it difficult to compare the efficacy of DG methods to the more traditional finite volume methods for industrial flow problems.

1.3. Thesis Overview

The rest of the thesis is structured as follows.

- Chapter 2 first defines the basis functions, and then derives discrete bilinear and linear operators (a and b in Eq. 1.8) from the governing equations and the boundary conditions.
- Chapter 3 treats the temporal discretization, which is based on a pressure correction method.
- Chapter 4 discusses several issues with the temporal discretization of variable-density flows.
- Chapter 5 contains simulations of turbulent plane channel flow, some of which were performed with an LES model.

There is little relation between the chapters, and each can be read independently.

References

[1] J. S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods* (Springer, New York, 2008).

[2] S. Paolucci, *On the Filtering of Sound from the Navier-Stokes Equations*, Tech. Rep. SAND82-8257 (Sandia National Laboratories, Ananlytical Thermal/Fluid Mechanics Division, Livermore, 1982).

[3] J. W. R. Peeters, *Turbulence and turbulent heat transfer at supercritical pressure*, doctoral thesis, Delft University of Technology (2016).

[4] F. C. Nicoud, *Numerical study of a channel flow with variable properties,* in *Annual Research Briefs* (Center for Turbulence Research, 1998) pp. 289–310.

[5] M. Hassanaly, H. Koo, C. F. Lietz, S. T. Chong, and V. Raman, *A minimally-dissipative low-Mach number solver for complex reacting flows in OpenFOAM,* Computers & Fluids **162**, 11 (2018).

[6] B. Müller, *Low-Mach-Number Asymptotics of the Navier-Stokes Equations,* Journal of Engineering Mathematics **34**, 97 (1998).

[7] W. H. Reed and T. R. Hill, *Triangular mesh methods for the neutron transport equation,* in *Proceedings of the American Nuclear Society* (University of California, Los Alamos Scientific Laboratory, Los Alamos, New Mexico 87544, 1973).

[8] S. S. Collis, *Discontinuous Galerkin methods for turbulence simulation,* in *Summer Program 2002, Center for Turbulence Research* (2002) pp. 155–167.

[9] A. Ern and J.-L. Guermond, *Discontinuous Galerkin Methods for Friedrichs' Systems,* in *Numerical Mathematics and Advanced Applications*, edited by A. B. de Castro, D. Gómez, P. Quintela, and P. Salgado (Springer Berlin Heidelberg, Berlin, Heidelberg, 2006) pp. 79–96.

[10] Y. Bazilevs and T. Hughes, *Weak imposition of Dirichlet boundary conditions in fluid mechanics,* Computers & Fluids **36**, 12 (2007).

[11] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini, *Unified analysis of discontinuous Galerkin methods for elliptic problems,* SIAM Journal on Numerical Analysis **39**, 1749 (2002).

[12] K. Shahbazi, *An explicit expression for the penalty parameter of the interior penalty method,* Journal of Computational Physics **205**, 401 (2005).

[13] K. Shahbazi, P. F. Fischer, and C. R. Ethier, *A high-order discontinuous Galerkin method for the unsteady incompressible Navier-Stokes equations,* Journal of Computational Physics **222**, 391 (2007).

[14] S. S. Collis, *The DG/VMS Method for Unified Turbulence Simulation,* in *32nd AIAA Fluid Dynamics Conference and Exhibit* (Houston, Texas, 2002).

[15] T. J. R. Hughes, G. Scovazzi, P. B. Bochev, and A. Buffa, *A multiscale discontinuous Galerkin method with the computational structure of a continuous Galerkin method,* Computer Methods in Applied Mechanics and Engineering **195**, 2761 (2006).

[16] M. Drosson, *Development of a high-order interior penalty discontinuous Galerkin method for compressible turbulent flows*, Phd thesis, University of Liège (2013).

[17] J.-B. Chapelier, M. De La Llave Plata, F. Renac, and E. Lamballais, *DNS of Canonical Turbulent Flows Using the Modal Discontinuous Galerkin Method,* in *Direct and Large-Eddy Simulation IX*, edited by J. Fröhlich, H. Kuerten, B. J. Geurts, and V. Armenio (Springer International Publishing, Dresden, 2015) pp. 91–96.

[18] H. L. Atkins and B. T. Helenbrook, *Numerical Evaluation of P-Multigrid Method for the Solution of Discontinuous Galerkin Discretizations of Diffusive Equations,* in *17th AIAA Computational Fluid Dynamics Conference* (Toronto, Ontario, 2005).

[19] B. Helenbrook and H. Atkins, *Application of P-Multigrid to Discontinuous Galerkin Formulations of the Poisson Equation,* in *17th AIAA Computational Fluid Dynamics Conference* (Toronto, Ontario, 2005).

[20] A. Abbà, L. Bonaventura, M. Nini, and M. Restelli, *Dynamic models for Large Eddy Simulation of compressible flows with a high order DG method,* Computers & Fluids **122**, 209 (2015), arXiv:1407.6591 .

[21] K. Hillewaert, *Development of the discontinuous Galerkin method for high-resolution, large scale CFD and acoustics in industrial geometries*, Phd thesis, Université catholique de Louvain (2013).

[22] K. A. Cliffe, E. J. C. Hall, and P. Houston, *Adaptive Discontinuous Galerkin Methods for Eigenvalue Problems Arising in Incompressible Fluid Flows,* SIAM Journal on Scientific Computing **31**, 4607 (2010).

[23] J. J. W. Van der Vegt and H. Van der Ven, *Space-Time Discontinuous Galerkin Finite Element Method with Dynamic Grid Motion for Inviscid Compressible Flows: I. General Formulation,* Journal of Computational Physics **182**, 546 (2002).

[24] V. Ambati and O. Bokhove, *Space-time discontinuous galerkin discretization of rotating shallow water equations,* Journal of Computational Physics **225**, 1233 (2007).

[25] S. Rhebergen, B. Cockburn, and J. J. W. van der Vegt, *A space-time discontinuous Galerkin method for the incompressible Navier-Stokes equations,* Journal of Computational Physics **233**, 339 (2013).

[26] M. Tavelli and M. Dumbser, *A pressure-based semi-implicit space-time discontinuous Galerkin method on staggered unstructured meshes for the solution of the compressible Navier-Stokes equations at all Mach numbers,* Journal of Computational Physics **341**, 341 (2017).

[27] M. Alhawwary and Z. J. Wang, *Fourier analysis and evaluation of DG, FD and compact difference methods for conservation laws,* Journal of Computational Physics **373**, 835 (2018).

[28] R. Moura, S. Sherwin, and J. Peiró, *Linear dispersion-diffusion analysis and its application to under-resolved turbulence simulations using discontinuous Galerkin spectral/hp methods,* Journal of Computational Physics **298**, 695 (2015).

[29] Y. Xia, J. Lou, H. Luo, J. Edwards, and F. Mueller, *OpenACC acceleration of an unstructured CFD solver based on a reconstructed discontinuous Galerkin method for compressible flows,* International Journal for Numerical Methods in Fluids **78**, 123 (2015).

[30] M. Kronbichler, B. Krank, N. Fehn, S. Legat, and W. A. Wall, *A New High-Order Discontinuous Galerkin Solver for DNS and LES of Turbulent Incompressible Flow,* in *New Results in Numerical and Experimental Fluid Mechanics XI*, edited by A. Dillmann, G. Heller, E. Krämer, C. Wagner, S. Bansmer, R. Radespiel, and R. Semaan (Springer International Publishing, Cham, 2018) pp. 467–477.

2

Spatial Discretization with a Discontinuous Galerkin Method

2.1. Introduction

This chapter details the spatial discretization, which is based on the discontinuous Galerkin (DG) finite element method. To simplify the discussion, we consider the stationary transport equations

$$\nabla \cdot \mathbf{m} = 0 \,, \tag{2.1a}$$

$$\nabla \cdot (\mathbf{u}\ \mathbf{m}) = \nabla \cdot \tau - \nabla p + \mathbf{F} \,, \tag{2.1b}$$

$$\nabla \cdot (\mathbf{m}\ h) = \nabla \cdot \left(\frac{k}{c_p} \nabla h \right) + Q \,, \tag{2.1c}$$

and postpone including the time variable to chapters 3 and 4. The temporal discretization that is discussed there does not depend on the spatial discretization, and understanding this chapter is not essential to understanding the rest of the thesis.

2.1.1. Overview of DG discretizations

The discontinuous Galerkin method can be thought of as a high-order finite volume method, replacing the constant solution within an element with an arbitrarily rich solution space that is defined everywhere on the interior of the element. The resulting numerical solution is discontinuous at the element boundaries. Finite volume and discontinuous Galerkin methods have in common that they discretize the weak form, rather than the original partial differential equation. As a result, numerical

Parts of this chapter have been published in [1].

fluxes that were developed for one-dimensional finite volume schemes carry over naturally to the DG method, making it particularly suitable for hyperbolic systems (see, e.g., the review [2], and a unified analysis of finite volume and DG discretizations in [3]).

DG methods for the diffusion operator have matured more recently. The diffusion equation has an 'instant smoothing' property, meaning that it has a differentiable solution everywhere in the domain, even when there are discontinuities in the initial condition, the boundary conditions, the forcing term, or the diffusion parameter. Unsurprisingly, a discontinuous solution space is not the most natural choice for these problems. Nevertheless, several DG approaches have emerged. We mention two of them here. See the standard reference [4] for a clear and thorough analysis.

The first approach is to split the Laplace operator into two equations with first-order derivatives, resulting in two discretizations, which can then be merged. This so-called 'local DG' (LDG) method is straightforward, but it leads to a large stencil, where each element is coupled to the neighbors of its neighbors.

We avoid this by using another approach, namely the symmetric interior penalty (SIP) method. Interior penalty methods have a local stencil, where each element is only coupled to it direct neighbors. One could take the following heuristic viewpoint of this method. Think of each element as a separate domain in which a spectral method is used to solve the temperature diffusion equation. The elements are coupled by providing each other's boundary conditions, of which we need two at each interior face: one for each neighbor. One of these boundary conditions is natural to a finite element framework: the heat flux must be the same at both sides of the face. The other boundary condition is that the temperature must be continuous, but enforcing this strongly would break the discontinuous nature of the solution space. Therefore continuity is enforced weakly by penalizing the jumps of the solution at the faces.

This gives rise to a user-defined penalty parameter that must be 'large enough' in order for the discrete bilinear form to be coercive, which is a sufficient condition for stability. This has been seen as a disadvantage, but several authors (e.g., [5, 6]) have found useful estimates, and this has been extended to various types of elements and unstructured meshes [7].

2.1.2. Construction of the Solution Space

To construct the finite element solution space, the domain Ω with outward normal n is meshed into a set of non-overlapping elements $\mathcal{T}$. The boundaries of an element are called faces. Internal faces each have two neighbors; boundary faces have only one.

We adopt the following notation from di Pietro and Ern [8]. The set of internal faces is denoted by $\mathcal{F}^{\mathrm{i}}$. The set of faces that lie on the Dirichlet (resp. Neumann) boundary of the domain are denoted by $\mathcal{F}^{\mathrm{D}}$ (resp. $\mathcal{F}^{\mathrm{N}}$). Sets of multiple types of faces are sometimes abbreviated with the obvious notation $\mathcal{F}^{\mathrm{D,i}} := \mathcal{F}^{\mathrm{D}} \cup \mathcal{F}^{\mathrm{i}}$. The set of faces of an element $T \in \mathcal{T}$ is $\mathcal{F}_T$. Similarly, $\mathcal{T}_F$ is the set of neighbors of face F. Each internal face F has a normal vector $\mathbf{n}^F$ that points in an arbitrary but fixed

direction. On boundary faces, $\mathbf{n}^F$ coincides with the outward normal of Ω.

The solution space is spanned by the basis functions, each of which has support on one element, meaning that it is zero elsewhere. The numerical solution is piecewise continuous: it is continuous within each element, but discontinuous across the faces. These discontinuities are described by the jump and average operators, which are defined on internal faces as

$$[\![x]\!] \coloneqq x^+ - x^- \quad \text{and} \quad \{x\} \coloneqq \frac{1}{2}\left(x^- + x^+\right) , \tag{2.2}$$

where

$$x^{\pm}(\mathbf{r} \in F) \coloneqq \lim_{\epsilon \downarrow 0} x(\mathbf{r} \mp \epsilon\ \mathbf{n}^F) \tag{2.3}$$

indicate the values at the two sides. On vectors they act element-wise. On boundary elements, both the jump and the average are defined as the internal numerical value (i.e., the 'trace'). The reason for this convention at the boundary is that it allows for a concise notation of the element-wise application of the divergence theorem: for an arbitrary piecewise continuous vector $\mathbf{v}$,

$$\sum_{T\in\mathcal{T}} \int_T \nabla\cdot\mathbf{v} = \sum_{T\in\mathcal{T}} \int_{\partial T} \mathbf{v}\cdot\mathbf{n}^T = \sum_{F\in\mathcal{F}^{\mathrm{D,N,i}}} \int_F [\![\mathbf{v}]\!]\cdot\mathbf{n}^F , \tag{2.4}$$

where $\mathbf{n}^T$ is the outward normal of element T.

Modal vs. Nodal Basis Functions

Though arbitrary types of basis functions are theoretically possible, discontinuous Galerkin methods almost always use polynomials. An exception is wall function enrichment, in which the solution space of wall-bounded elements is enriched with functions that contain some physically motivated, a priori information on the flow near the wall flow (e.g., [9, 10]). The polynomials can be either *modal* or *nodal*.

Nodal functions are defined through nodes, which are coordinates such that the i^{th} basis function equals δ_{ij} on the j^{th} node, that is, the function is nonzero on exactly one of the nodes. This has a computational advantage when evaluating the integrals in the discrete weak forms: the nodes can be placed on the quadrature points, making the numerical quadrature a sparse sum. Many recent DG implementations use this approach to speed up the matrix assembly. The book by Hesthaven and Warburton [11] is a good introduction to this topic, and to the DG method in general.

Nodal functions are often defined such that they span a tensor product of one-dimensional polynomial spaces. For example, a second-order polynomial approximation on a two-dimensional element would contain all functions in $\{1, x, x^2\} \times \{1, y, y^2\} = \{1, y, y^2,\ x, xy, xy^2,\ x^2, x^2y, x^2y^2\}$. In this case a polynomial order $\mathcal{P}$ actually means that all polynomials of order less than or equal to $\mathcal{P}$ lie in the solution space, but it also contains some higher-order functions. There are $(\mathcal{P}+1)^d$ basis functions in each d-dimensional element. Note that the basis depends on the orientation of the coordinate axes.

We use a modal basis, in which the basis functions are constructed such that they are hierarchical and orthogonal in the L^2-norm. This is just an implementation issue; all equations in this thesis are valid for non-orthogonal bases. The solution space within each element is simply spanned by all polynomials up to an order $\mathcal{P}$. There are $\binom{\mathcal{P}}{\mathcal{P}+d} = (\mathcal{P}+d)!/(d!\,\mathcal{P}!)$ linearly independent polynomials in a d-dimensional element. In this chapter the polynomial order is the same on all elements, though this is not a requirement of the numerical method. The order of the polynomials for the unknown X is denoted by $\mathcal{P}_X$.

2.2. Discrete Continuity Equation

The DG bilinear form for the divergence operator can be found in many previous works, including [12], [13, p. 92], and [8, pp. 250–252], and is given by

$$a^{\mathrm{div}}(\mathbf{v}, q) = -\sum_{T\in\mathcal{T}} \int_T q\nabla\cdot\mathbf{v} + \sum_{F\in\mathcal{F}^{\mathrm{D,i}}} \int_F \{q\}\,[\![\mathbf{v}]\!]\cdot\mathbf{n}^F \;, \tag{2.5}$$

so that a consistent discrete weak form of the continuity equation with $\partial\rho/\partial t = 0$ (Eq. 2.1a) is

$$\begin{aligned} &\textit{Find } p \in V_p\textit{, such that, for all } q \in V_p\textit{,}\\ &-a^{\mathrm{div}}(\mathbf{m}, q) = -\int_{\partial\Omega^{\mathrm{D}}} q\;\mathbf{m}^{\mathrm{D}}\cdot\mathbf{n}\;, \end{aligned} \tag{2.6}$$

where V_p is the solution space of the pressure, and $\mathbf{m}$ is the numerical (discontinuous) mass flux.

The weak form of the continuity equation can only be stable if the divergence operator is surjective. This means that, for every q in the pressure space, there exists a $\mathbf{v}$ in the velocity (or mass flux) space, such that q is the divergence of $\mathbf{v}$ (and satisfying a constraint on the norm of $\mathbf{v}$; see [8, pp. 246–252] for a precise analysis). Surjectivity can be shown to be equivalent to an inf-sup condition on the bilinear form of the divergence operator.

The continuous divergence operator is surjective, but the discrete divergence operator a^{div} is not for equal-order discretizations of the pressure and the mass flux (i.e., $\mathcal{P}_m = \mathcal{P}_p$). This makes intuitive sense, since the divergence of the mass flux would lie in a lower-order polynomial space than the pressure. A solution is to set $\mathcal{P}_m > \mathcal{P}_p$, in which case inf-sup stability has been proven [14]. The lack of inf-sup stability is a general aspect of equal-order finite element methods for incompressible flows. See, for example, John [15] for an extensive discussion in the context of continuous finite elements.

This is unfortunate, because equal-order methods have often been found more efficient than mixed-order methods, and therefore methods have been devised to stabilize the pressure for equal-order DG discretizations. Cockburn *et al.* [16] assumed a homogeneous kinematic viscosity and added a pressure stabilization term,

so that Eq. 2.6 becomes

$$\begin{aligned} &\textit{Find } p \in V_p\textit{, such that, for all } q \in V_p\textit{,} \\ &-a^{\mathrm{div}}(\mathbf{m}, q) + a^{\mathrm{stab}}(p, q) = -\int_{\partial\Omega^{\mathrm{D}}} q\ \mathbf{m}^{\mathrm{D}} \cdot \mathbf{n}\ , \end{aligned} \tag{2.7}$$

where

$$a^{\mathrm{stab}}(p, q) = \sum_{F \in \mathcal{F}^i} \int_F \zeta^F \, [\![p]\!] \, [\![q]\!] \tag{2.8}$$

for equal-order discretizations, and $a^{\mathrm{stab}}(p, q) = 0$ for mixed-order discretizations. The penalty parameter for the pressure discontinuities is $\zeta^F = \gamma_0 \, \|F\|_{\mathrm{leb}} / \nu$, where $\|\cdot\|_{\mathrm{leb}}$ is the Lebesgue measure (which is the length, area, or volume in 1, 2, or 3 dimensions). We adjust the above penalty term to a variable viscosity in the obvious way: by taking the pointwise maximum value of ζ^F on both sides of the face, that is,

$$\zeta^F = \gamma_0 \, \|F\|_{\mathrm{leb}} \max_{T \in \mathcal{T}_F} \left[\frac{1}{\nu}\right]_T \ . \tag{2.9}$$

We set $\gamma_0 = 1$ without investigating other values. We offer no proof for the validity of this handling of the variable viscosity, but extensive tests in this thesis (especially those with manufactured solutions in sections 3.3.2 and 4.5.1) will show that the discretization is stable.

Another approach to stabilizing equal-order DG methods was taken by Botti and Di Pietro [17], who used continuous finite elements for the pressure, simply disposing of the pressure discontinuities altogether. This seems logical, since the transport equations for an incompressible flow imply a Poisson equation for the pressure, and the continuous Galerkin (CG) method is the most effective discretization for purely diffusive problems, whereas the DG method mostly thrives for hyperbolic problems. (Though it has been shown that DG and CG methods are asymptotically equally efficient in the limit of (very) high polynomial orders of approximation [18].) A possible disadvantage would be the reduced mesh generality compared to a pure DG discretization. We have not pursued this approach, partly because it would be demanding to implement.

Krank *et al.* [19] have instead focused on the momentum equation to stabilize equal-order DG discretizations. They suppressed the local violation of the continuity equation by modifying the discrete Navier-Stokes equation with element-wise penalty terms for $\nabla \cdot \mathbf{u}$ within an element, and for the jump of $\mathbf{n} \cdot \mathbf{m}$ across the faces. This assumes a divergence-free velocity field, which is generally not valid for low-Mach number flow. We will use this approach in chapter 5.

These ideas for pressure stabilization can be compared to the artificial compressibility method, in which the continuity equation is perturbed with a compressibility term $(1/c^2)\partial p/\partial t$ for some parameter $c > 0$. The numerical fluxes in the DG weak form are then obtained by solving a Riemann problem for the discontinuities at a face. This also gives rise to penalty terms for the jump of the pressure (as in Eq. 2.8), and the jump of $\mathbf{n} \cdot \mathbf{m}$ across a face (ref., e.g, [20]).

2.3. Discrete Momentum Equation

The discrete weak form of the stationary momentum equation (Eq. 2.1b) is

Find $\mathbf{m} \in V_m$*, such that, for all* $\mathbf{v} \in V_m$*,*

$$a^{\text{conv}}(\mathbf{u}, m_k, v_k) + a^{\text{visc}}(\mathbf{m}, \mathbf{v}) = l^{\text{conv}}(\mathbf{u}, m_k^{\text{D}}, v_k) + l^{\text{visc}}(\mathbf{v}) - a^{\text{div}}(\mathbf{v}, p) + \int_\Omega \mathbf{F} \cdot \mathbf{v} \,, \tag{2.10}$$

where V_m is the solution space of the mass flux, and p is the numerical (discontinuous) pressure. Note that the divergence operator a^{div} doubles as a gradient operator. Integrating Eq. 2.5 by parts, and using the fact that $[\![q\mathbf{v}]\!] = [\![q]\!]\{\mathbf{v}\} + \{q\}[\![\mathbf{v}]\!]$ on an interior face, gives

$$a^{\text{div}}(\mathbf{v}, q) = \sum_{T \in \mathcal{T}} \int_T \mathbf{v} \cdot \nabla q - \sum_{F \in \mathcal{F}^{\text{N,i}}} \int_F [\![q]\!]\{\mathbf{v}\} \cdot \mathbf{n}^F \,, \tag{2.11}$$

demonstrating consistency, since the last term is zero when the continuous pressure is substituted for q.

The discretization of the convection a^{conv} and l^{conv} will be given in section 2.5. Solving for the mass flux $\mathbf{m}$ instead of the velocity $\mathbf{u}$ complicates the treatment of the viscous term, which is linear in $\nabla\mathbf{u}$, not $\nabla\mathbf{m}$. Section 2.3.1 details how this can be handled with a DG method.

2.3.1. Discretization of the Viscous Stress

To derive a discretization for the viscous term, rewrite the viscous stress in terms of the mass flux as $\tau = L^{\text{visc}}(m)$, where

$$L_{ij}^{\text{visc}}(\mathbf{m}) = \frac{\mu}{\rho}\left(B_{ij} + B_{ji} - \frac{2}{3}B_{kk}\delta_{ij}\right) \,, \tag{2.12}$$

is a linear operator, with $B_{ij} = \rho\nabla_i u_j = \nabla_i m_j - d_i m_j$, and

$$\mathbf{d} \coloneqq \frac{1}{\rho}\nabla\rho \,. \tag{2.13}$$

We use a generalization of the symmetric interior penalty (SIP) method, given by the discrete bilinear operator

$$\begin{aligned} a^{\text{visc}}(\mathbf{w}, \mathbf{v}) = & \sum_{T \in \mathcal{T}} \int_T L_{kl}^{\text{visc}}(\mathbf{w})\, \nabla_k v_l + \sum_{F \in \mathcal{F}^{\text{i}} \cup \mathcal{F}^{\text{D}}} \int_F \eta^F [\![\mathbf{w}]\!] \cdot [\![\mathbf{v}]\!] \\ & - \sum_{F \in \mathcal{F}^{\text{i}} \cup \mathcal{F}^{\text{D}}} \int_F \left([\![\mathbf{v}]\!] \cdot \left\{L^{\text{visc}}(\mathbf{w})\right\} + [\![\mathbf{w}]\!] \cdot \left\{L^{\text{visc}}(\mathbf{v})\right\}\right) \cdot \mathbf{n}^F \end{aligned} \tag{2.14}$$

and the linear operator

$$l^{\text{visc}}(\mathbf{v}) = \sum_{F \in \mathcal{F}^{\text{D}}} \int_F \left(\eta^F\, \mathbf{m}^{\text{D}} \cdot \mathbf{v} - \mathbf{m}^{\text{D}} \cdot L^{\text{visc}}(\mathbf{v}) \cdot \mathbf{n}^F\right) + \int_{\partial\Omega^{\text{N}}} \mathbf{f}^{\text{N}} \cdot \mathbf{v} \,. \tag{2.15}$$

This reduces to the regular SIP method when substituting $(\mu/\rho)\nabla_i w_j$ for $L_{ij}^{\text{visc}}(w)$. Compared to other interior penalty methods and the local DG method, the advantages of the SIP method are the optimal convergence rate for all polynomial orders, its adjoint consistency, and its compact stencil [21].

If the viscosity is constant and the flow is incompressible (i.e., $\nabla\cdot\mathbf{u} = 0$), then $\nabla\cdot\tau$ can be simplified to $\nabla\cdot(\mu\nabla\mathbf{u})$ on the continuous level. But note that using a regular SIP method for $\nabla\cdot(\mu\nabla\mathbf{u})$ would always be different from the above discretization of $\nabla\cdot\tau$. As one would expect, our numerical tests (not shown in this thesis) show a negligible numerical difference between these discretizations when μ is constant and $\nabla\cdot\mathbf{u} = 0$.

The above discretization of the viscous term can be compared to what is usually done for compressible flows, where the system of equations 1.1 is solved for a full state vector $U \coloneqq [\rho, \mathbf{m}, \rho h]$. In that case all elliptic terms in Eqs. 1.1 can be written as $\nabla\cdot(G(U)\nabla U))$, where $G(U)$ is a homogeneity tensor that does not contain any gradients of the unknowns. (See, e.g., [21, 22].) This tensor is then kept fixed during an iteration step, while ∇U can be treated in a time-implicit manner. If the density is constant, then that approach is equivalent to the above discretization.

If the density is variable (as it will be in later chapters), then using a homogeneity tensor is subtly different from the current method in terms of which variables get treated implicitly. Three of the six terms in Eq. 2.12 contain $\mathbf{d}\,\mathbf{m}$, which is a product of $\mathbf{m}$, $1/\rho$, and $\nabla\rho$. When using a homogeneity tensor, $\mathbf{m}$ and $(1/\rho)$ are frozen within an iteration step, and $\nabla\rho$ is solved for implicitly. In the current discretization, $(1/\rho)$ and $\nabla\rho$ are frozen, and $\mathbf{m}$ is treated implicitly.

Our approach also differs from that of Klein *et al.* [23], in that we treat all terms in the viscous stress (Eq. 1.2) in a time-implicit manner, whereas they only do this for the first term $(\mu\nabla_i u_j)$. In our treatment the velocity components are coupled. We have no a priori estimate for the difference in magnitude between the effects of the first term $(\mu\nabla_i u_j)$ and its transpose $(\mu\nabla_j u_i)$ on the viscous force $\nabla\cdot\tau$, especially when the viscosity varies strongly in space. Note the gradients in the effective viscosity will increase greatly when a large eddy simulation (LES) model is included in chapter 5.

Following Hillewaert [7, p. 30], we set the penalty parameter to

$$\eta^F = \max_{T\in\mathcal{T}_F}\left(C_T\ \text{card}(\mathcal{F}_T)\ \frac{\|F\|_{\text{leb}}}{\|T\|_{\text{leb}}}\right)\ \max_{T\in\mathcal{T}_F}(K|_T)\ , \tag{2.16}$$

where $K = \mu/\rho$ is the diffusion parameter, and $\text{card}(\mathcal{F}_T)$ is the number of faces of element T. The factor C_T depends on the type of elements in the mesh: for a polynomial order $\mathcal{P}$, $C_T = (\mathcal{P}+1)^2$ for quadrilaterals and hexahedra, $C_T = (\mathcal{P}+1)(\mathcal{P}+2)/2$ on triangles, and $C_T = (\mathcal{P}+1)(\mathcal{P}+3)/3$ for tetrahedra.

We compute the penalty parameter in a pointwise manner, even though Hillewaert took the maximum value of the above expression on the face in his stability analysis. Our experience suggests no difference in the stability, and taking a local, pointwise (as opposed to face-averaged) numerical flux seems more in the spirit of the DG method, which can have an arbitrarily rich structure within an element. We

will encounter a similar situation for the stabilization term for the local Lax-Friedrichs flux in section 2.5.1.

Paradoxically Eq. 2.16 depends on the shape, size, and number of faces of the neighbors, giving the impression that it is still not local to a particular point on a face, even though we evaluate it in a pointwise manner. However, the norms of a polynomial on a face and on an element are related by well-known trace inequalities, which underlie the above expression for the penalty parameter.

2.4. Discrete Enthalpy Equation

We solve for the specific enthalpy h from the enthalpy transport equation in conservative form. Given a solution space V_h for the enthalpy, the discrete weak form of the stationary enthalpy equation 2.1c is

$$\begin{aligned} &\textit{Find } h \in V_h\textit{, such that, for all } v \in V_h\textit{,} \\ &a^{\mathrm{conv}}(\mathbf{m}, h, v) + a^{\mathrm{SIP}}(h, v) = l^{\mathrm{conv}}(\mathbf{m}, h^{\mathrm{D}}, v) + l^{\mathrm{SIP}}(v) + \int_\Omega Q\ v\ , \end{aligned} \tag{2.17}$$

where a^{SIP} and l^{SIP} are standard SIP bilinear and linear forms to discretize the Fourier heat flux. The SIP penalty parameter is as in Eq. 2.16, with a diffusion coefficient $K = k/c_p$. Note that the convective discretization is the same as for the mass flux, except that the convecting field is $\mathbf{m}$, rather than $\mathbf{u} = (1/\rho)\mathbf{m}$. (That is, $\mathbf{b} = \mathbf{m}$ in Eqs. 2.20–2.21.) This is convenient, because we also solve for $\mathbf{m}$.

2.5. Discretization of the Convection

To derive an expression for the convective bilinear and linear forms (a^{conv} and l^{conv} in Eqs. 2.10 and 2.17), consider the time-independent, purely convective problem with a numerical (discontinuous) advecting field $\mathbf{b}$, and an unknown generic scalar ϕ:

$$\nabla \cdot (\mathbf{b}\ \phi) = 0\ . \tag{2.18}$$

Though this equation is linear (because $\mathbf{b}$ is given), its analysis is also relevant to the nonlinear convection in the momentum equation, which must be linearized with some estimate for the convecting velocity field.

At $\partial\Omega^{\mathrm{D}}$ we have the Dirichlet values $\mathbf{b}^{\mathrm{D}} \cdot \mathbf{n}$ and ϕ^{D}, whereas there may be no numerical inflow (i.e., $\mathbf{b} \cdot \mathbf{n} < 0$) at the Neumann boundary. Note that ϕ^{D} is only defined where $\mathbf{b}^{\mathrm{D}} \cdot \mathbf{n} < 0$. This is an important point to which we will return later: the inflow region is defined by the known value $\mathbf{b}^{\mathrm{D}} \cdot \mathbf{n}$, not by the sign of the numerical value $\mathbf{b} \cdot \mathbf{n}$.

The discrete weak form of Eq. 2.18 is

$$\begin{aligned} &\textit{Find } \phi \in V_\phi\textit{, such that, for all } v \in V_\phi\textit{,} \\ &a^{\mathrm{conv}}(\mathbf{b}, \phi, v) = l^{\mathrm{conv}}(\mathbf{b}, \phi^{\mathrm{D}}, v)\ , \end{aligned} \tag{2.19}$$

where V_ϕ is the solution space of ϕ. The linear term is

$$l^{\mathrm{conv}}(\mathbf{b}, \phi^{\mathrm{D}}, v) = -\int_{\partial\Omega^{\mathrm{D}}} v\ \min\left(0, \mathbf{b}^{\mathrm{D}} \cdot \mathbf{n}\right) \phi^{\mathrm{D}}\ . \tag{2.20}$$

The bilinear term has the general form

$$a^{\text{conv}}(\mathbf{b}, w, v) = -\sum_{T\in\mathcal{T}}\int_T w\ \mathbf{b}\cdot\nabla v + \sum_{F\in\mathcal{F}^{\text{i}}}\int_F [\![v]\!]\, H^F(\mathbf{b}, w) + \int_{\partial\Omega^{\text{N}}} v\ w\ \mathbf{b}\cdot\mathbf{n} + \int_{\partial\Omega^{\text{D}}} v\ w \max\left(0, \mathbf{b}^{\text{D}}\cdot\mathbf{n}\right)\ . \tag{2.21}$$

Here w is a scalar, v is the test function, $\mathbf{b}$ is the convective field, and H^F is the numerical flux function on a face F, which will be defined later.

It is well known that imposing a Dirichlet boundary condition for the velocity at an outlet results in an ill-posed problem, and that it is numerically unstable for a convection-dominated flow, and therefore we would normally have $\max(0, \mathbf{n}\cdot\mathbf{b}^{\text{D}}) = 0$ on $\partial\Omega^{\text{D}}$. Here we nevertheless include this term in a^{conv}, because we will use it in the Taylor-Green vortex in section 3.3.1, as is standard practice for that laminar benchmark case (e.g., [12, 24]).

In practice one may know the value ϕ^{D} at a Dirichlet outlet ($\mathbf{n}\cdot\mathbf{b}^{\text{D}} > 0$), such as for the Taylor-Green vortex manufactured solution in section 3.3.1, but we nonetheless use the internal value w in the last term in Eq. 2.21, so as not to overconstrain the problem. This theoretical point has little practical value, because Dirichlet outlets are only viable for academic problems anyway.

If (the normal component of) b is continuous at each face, then ϕ can be upwinded in an unambiguous manner, and there is only one correct discretization (see, e.g., [21, p. 33]). In practice $\mathbf{b}$ is a velocity (or a mass flux) that was obtained with a DG method, and therefore the flux is discontinuous at each face, and there are multiple possible discretizations.

2.5.1. Choice of the Numerical Flux

The convective discretization is closed by defining a numerical flux function H^F in Eq. 2.21 for an internal face F. It only depends on the basis function (w) and the normal component of the flow ($\mathbf{b}\cdot\mathbf{n}$) on both sides of the face; there is no interpolation between the elements. The numerical flux can therefore be borrowed from finite volume methods for one-dimensional hyperbolic conservation laws. There are many monographs on this subject, including the standard works by LeVeque [25, 26]. Toro [27] offers a particularly clear overview with an emphasis on the Euler equations.

The most principled numerical flux can be obtained by solving the associated Riemann problem at the discontinuity exactly, which is known as Godunov's method. See, for example, [8, p. 105] and especially [3] for notes in a DG context. Godunov's method is known to lead to the least amount of numerical dissipation [2, 28]. Unfortunately the Riemann problem typically takes up a substantial part of the total computation time. It is therefore more common to use an approximation for the numerical flux.

We use the local Lax-Friedrichs flux, which is cheap, but also known to be fairly dissipative. Among the many other numerical fluxes, we mention the following.

- The Roe numerical flux, which has been compared favorably to the local Lax-Friedrichs flux by Winters *et al.* [29]. They focused on the interplay between the numerical flux and their under-integrated nodal DG method that required anti-aliasing, and so it is hard to say how well their results would translate to our modal DG method with exact integration in the weak forms.
- The Vijayasundaram numerical flux. See, for example, the seminal paper [30] for a general analysis, and [31] for a DG application. In a DG method this numerical flux can be shown to be consistent with the numerical flux in the continuity equation.

The Vijayasundaram flux has also been implemented in `DGFlows`; our experience suggests no significant difference with respect to the local Lax-Friedrichs flux.

The local Lax-Friedrichs flux is given by

$$H^F(\mathbf{b}, w) = \frac{1}{2}[\![w]\!]\,\alpha^F + \{w\mathbf{b}\}\cdot n^F\ , \tag{2.22}$$

where α^F is a function of the normal component of the convecting field (i.e., $\mathbf{b}\cdot\mathbf{n}$). For a general scalar hyperbolic system of the form $\nabla\cdot\mathbf{f}(\phi)=0$ with a given vector-valued function $\mathbf{f}$, the parameter α^F is the maximum value of $\left|\mathbf{f}'(\phi)\cdot\mathbf{n}^F\right|$ on either side of the face. If ϕ is an advected scalar, such as for the enthalpy in Eq. 2.17, then $\mathbf{f}(w)=w\,\mathbf{b}$, and thus $\left|\mathbf{f}'(\phi)\cdot\mathbf{n}^F\right| = \left|\mathbf{b}\cdot\mathbf{n}^F\right|$.

The situation is different for the nonlinear convection in the momentum equation 2.10, where ϕ is itself equal to the convecting field $\mathbf{b}$ (up to a factor of the density). In that case, $\phi=\mathbf{m}$, $\mathbf{b}=\mathbf{u}=(1/\rho)\mathbf{m}$, and Eq. 2.18 becomes a vector equation of the form $\nabla\cdot F(\mathbf{m})=0$ with $F_{ij}(\mathbf{w}) = w_i w_j/\rho$. Now α^F is the maximum of the spectral radii of the Jacobi matrix $\partial/\partial\phi_i\left(F_{jk}(\phi)n_k^F\right) = \partial/\partial m_i\left(m_j\left(\mathbf{m}\cdot\mathbf{n}^F\right)\right)/\rho = \left(\mathbf{u}\cdot\mathbf{n}^F\right)\delta_{ij} + n_i^F u_j$ on either side of the face. Its eigenvectors are either parallel to $\mathbf{n}$ (with eigenvalue $2(\mathbf{u}\cdot\mathbf{n}^F)$), or perpendicular to $\mathbf{u}$ (with eigenvalue $(\mathbf{u}\cdot\mathbf{n}^F)$). This can be summarized as

$$\alpha^F = \varkappa \max_{T\in\mathcal{T}_F}\left|\mathbf{b}\cdot\mathbf{n}^F\right|_T \tag{2.23}$$

with $\varkappa=1$ for advected scalars (such as the enthalpy in Eq. 2.17), and $\varkappa=2$ for the nonlinear convection in the momentum equation 2.10 (see, e.g., [2, 12]).

Note that we evaluate α^F in a pointwise manner in the integral in Eq. 2.21, which is the only right choice if ϕ is an advected scalar. Averaging α^F over the face would be inconsistent, as can be shown by considering the special case of a continuous convecting field $\tilde{\mathbf{b}}$ (i.e., $[\![\tilde{\mathbf{b}}]\!]=0$ on internal faces, and $\tilde{\mathbf{b}}=\mathbf{b}^{\mathrm{D}}$ on $\partial\Omega^{\mathrm{D}}$). With a test function that is $v=1$ on an element T and $v=0$ elsewhere, the discretization Eq. 2.19 can be written as

$$\begin{aligned} 0 = &\int_{\partial T\cap\partial\Omega^{\mathrm{N}}} w\,(\tilde{\mathbf{b}}\cdot\mathbf{n}) + \int_{\partial T\cap\partial\Omega^{\mathrm{D}}} w^{\mathrm{up}}\left(\mathbf{b}^{\mathrm{D}}\cdot\mathbf{n}\right) \\ &+ \sum_{F\in\mathcal{F}_T\cap\mathcal{F}^{\mathrm{i}}}\int_F \frac{1}{2}\left(\left(w^+ - w^-\right)\alpha^F + \left(w^+ + w^-\right)\tilde{\mathbf{b}}\cdot\mathbf{n}^T\right)\ , \end{aligned} \tag{2.24}$$

where $\mathbf{n}^T$ is the outward normal of element T, w^+ is the value on T, w^- is the value on its neighbor, and

$$w^{\mathrm{up}} := \begin{cases} \phi^{\mathrm{D}}\,, & \text{for } \mathbf{n}\cdot\mathbf{b}^{\mathrm{D}} < 0, \\ w\,, & \text{for } \mathbf{n}\cdot\mathbf{b}^{\mathrm{D}} > 0 \end{cases} \tag{2.25}$$

is the upstream value at a Dirichlet boundary. Since $\tilde{\mathbf{b}}$ is continuous, this must reduce to a pure upwind flux, so the last integrand must equal $\left(\tilde{\mathbf{b}}\cdot\mathbf{n}^T\right) w^{\pm}$ for $\pm\tilde{\mathbf{b}}\cdot\mathbf{n}^T > 0$. This only holds if $\alpha^F = \left|\tilde{\mathbf{b}}\cdot\mathbf{n}^T\right|$ everywhere on F.[1]

The above argument breaks down for the nonlinear convection term in the momentum equation 2.10, because then ϕ and $\mathbf{b}$ are equivalent, and so a continuous $\mathbf{b}$ would imply a continuous ϕ. Nevertheless our experience suggests that evaluating α^F in a pointwise manner sometimes results in noticeably lower errors, especially for high-order numerical solutions, whereas the stability does not seem negatively impacted. As mentioned in section 2.3.1, averaging a penalty parameter over a face or an element runs counter to our intuitive understanding of a high-order DG method.

This opinion does not appear universally shared in the literature. Cockburn and Shu [2] have suggested using to the two element averages on the neighbors to compute α^F for the nonlinear convection term. Shahbazi *et al.* [12] was perhaps the first to put this into practice, and others have followed (e.g., [32]). More recently, de la Llave Plata *et al.* [33] have evaluated α^F in a pointwise manner with underresolved DG large eddy simulations, and Tavelli and Dumbser [34] have done the same for a space-time DG method, where averaging over a space-time element would presumably have been more involved. None of the above authors have given explicit reasons for their choices, and we are not aware of a systematic study on averaging α^F or not.

2.5.2. Solution Spaces for the Enthalpy and the Pressure

If a scalar quantity ϕ is advected with a velocity field that was obtained with a discontinuous Galerkin method, then the solution space of the scalar must be a subset of the solution space of the pressure, that is, $\mathcal{P}_\phi \leq \mathcal{P}_p$. Discretizations with $\mathcal{P}_\phi > \mathcal{P}_p$ are inconsistent, and therefore often unstable, because the continuity equation is weighed by the pressure basis functions, so that the numerical velocity only satisfies the incompressibility constraint in a weak sense up to order $\mathcal{P}_p$. This means that the convective discretization can only be consistent up to an order $\mathcal{P}_p$ [1, 35].

To make this more precise, consider the discrete continuity equation in Eq. 2.7 with the divergence operator in the form of Eq. 2.5:

$$-a^{\mathrm{stab}}(p,q) + \sum_{T\in\mathcal{T}_h}\int_T \mathbf{m}\cdot\nabla q = \sum_{F\in\mathcal{F}^{\mathrm{i}}}\int_F [\![q]\!]\,\{\mathbf{m}\}\cdot\mathbf{n}^F + \int_{\partial\Omega^{\mathrm{N}}} q\,\mathbf{m}\cdot\mathbf{n} + \int_{\partial\Omega^{\mathrm{D}}} q\,\mathbf{m}^{\mathrm{D}}\cdot\mathbf{n}\,, \tag{2.26}$$

for a test function q that lies in the pressure solution space. This can be compared to the advection discretization in Eq. 2.19 by substituting the continuous solution

[1] This is also another way of showing that $\varkappa = 1$ for a scalar ϕ.

$\phi \leftarrow \tilde{\phi}$, which satisfies $[\![\tilde{\phi}]\!] = 0$ on internal faces and $\tilde{\phi} = \phi^{\mathrm{D}}$ on $\partial\Omega^{\mathrm{D}}$. Every viable numerical flux must reduce to the central flux for a continuous solution, that is, $H^F\left(\mathbf{b}, \tilde{\phi}\right) = \tilde{\phi}\,\{\mathbf{b}\} \cdot \mathbf{n}^F$ (see, e.g., [21]), so that Eqs. 2.19–2.21 become

$$\sum_{T\in\mathcal{T}} \int_T \tilde{\phi}\ \mathbf{b} \cdot \nabla v = \sum_{F\in\mathcal{F}^{\mathrm{i}}} \int_F \tilde{\phi}\,[\![v]\!]\,\{\mathbf{b}\} \cdot \mathbf{n}^F + \int_{\partial\Omega^{\mathrm{N}}} \tilde{\phi}\ v\ \mathbf{b} \cdot \mathbf{n} + \int_{\partial\Omega^{\mathrm{D}}} \phi^{\mathrm{D}} v\ \mathbf{b}^{\mathrm{D}} \cdot \mathbf{n}\ . \quad (2.27)$$

This can clearly only be consistent for a test function v that is part of the test space of the continuity equation. In the special case of a constant solution $\tilde{\phi}$, the convective term should vanish for all v, but $a^{\mathrm{conv}}(\mathbf{m}, \tilde{\phi}, v) = \tilde{\phi}\ a^{\mathrm{div}}(\mathbf{m}, v)$, which only vanishes if the test function v lies in the pressure solution space, that is, if $\mathcal{P}_\phi \leq \mathcal{P}_p$.

The requirement $\mathcal{P}_\phi \leq \mathcal{P}_p$ was not satisfied in some previous literature on mixed-order DG schemes. For example, Klein *et al.* [23] chose the same solution space for the temperature as for the components of the velocity field. They probably found good results because their tests were done at a low Prandtl number of 0.7, whereas the problem with the solution spaces manifests itself when the convective term dominates.

In theory the discretization of the convection of a scalar is consistent as long as its solution space is a subset of the solution space of the pressure, but in practice the enthalpy has the same solution space as the pressure, because a scalar transport equation is much cheaper to solve than the momentum and pressure equations, so there is little reason not to obtain the highest available spatial accuracy for the enthalpy. In later chapters that address low-Mach number flow, the density will be a function of the temperature (and thus, the enthalpy). Then the choice of the enthalpy solution space is no longer free: it *must* be the same as that of the pressure, because the continuity equation and the enthalpy equation become coupled due to the temperature-dependent density.

2.5.3. Proper Treatment of Dirichlet Boundary Conditions

Regardless of the numerical flux, setting $v = 1$ in the discretization in Eqs. 2.19–2.21 reveals the global conservation property

$$0 = \int_{\partial\Omega^{\mathrm{N}}} w\ \mathbf{b} \cdot \mathbf{n} + \int_{\partial\Omega^{\mathrm{D}}} w^{\mathrm{up}}\ \mathbf{b}^{\mathrm{D}} \cdot \mathbf{n} \quad (2.28)$$

with w^{up} as in Eq. 2.25, as opposed to $\int_{\partial\Omega} w\ \mathbf{b} \cdot \mathbf{n} = 0$, which one may have guessed from the governing equation 2.18. This shows the effect of our treatment of the Dirichlet boundary condition: the total inflow is determined by the known boundary conditions (i.e., $(\phi\mathbf{b})^{\mathrm{D}} \cdot \mathbf{n}$), rather than the internal value (i.e., $\phi\ \mathbf{b} \cdot \mathbf{n}$), which generally contains a numerical error.

Another way of looking at this is to rewrite Eq. 2.27 for the continuous solution $\tilde{\phi}$ as

$$\sum_{T\in\mathcal{T}} \int_T \tilde{\phi}\ \mathbf{b} \cdot \nabla v - \sum_{F\in\mathcal{F}^{\mathrm{i,N,D}}} \int_F \tilde{\phi}\,[\![v]\!]\,\{\mathbf{b}\} \cdot \mathbf{n}^F = \int_{\partial\Omega^{\mathrm{D}}} \phi^{\mathrm{D}} v\left(\mathbf{b}^{\mathrm{D}} - \mathbf{b}\right) \cdot \mathbf{n}\ . \quad (2.29)$$

The right-hand side acts as a source to counterbalance the deviation of the numerical inflow from the imposed inflow at the boundary. Note that the inflow term in the discrete continuity equation 2.7 must be consistent with the treatment of the Dirichlet boundary in the convection discretization. Otherwise Eqs. 2.26 and 2.27 in the previous section would not have been consistent in the special case of a constant continuous solution $\tilde{\phi}$.

Eqs. 2.20–2.21 present the only correct treatment of the Dirichlet boundary, though it is far from standard in recent literature. For example, Piatkowski *et al.* [31] used a Vijayasundaram numerical flux $H^F(\mathbf{b}, w) = w^+ \max\left(0, \{\mathbf{b}\} \cdot \mathbf{n}^F\right) + w^- \min\left(0, \{\mathbf{b}\} \cdot \mathbf{n}^F\right)$ on internal faces, and extended this in a seemingly logical way to Dirichlet faces by replacing w^- with ϕ^D, so that the contribution of the Dirichlet boundary became

$$\int_{\partial\Omega^\mathrm{D}} v \left(w \max\left(0, \mathbf{b} \cdot \mathbf{n}\right) + \phi^D \min\left(0, \mathbf{b} \cdot \mathbf{n}\right) \right) . \tag{2.30}$$

This is subtly wrong, because the numerical value $\mathbf{b} \cdot \mathbf{n}$ may have any sign, though a value for ϕ^D should not be required at outlets and walls (where $\mathbf{b}^\mathrm{D} \cdot \mathbf{n} = 0$).

A similar objection can be made to the boundary treatment in Shahbazi *et al.* [12], who used a local Lax-Friedrichs flux as in Eq. 2.22, and substituted $\{w\ \mathbf{b}\} \longleftarrow \left(\phi\mathbf{b} + (\phi\ \mathbf{b})^\mathrm{D}\right)/2$ and $[\![w]\!] \longleftarrow w - \phi^\mathrm{D}$ on $\partial\Omega^\mathrm{D}$, giving a Dirichlet boundary contribution

$$\int_{\partial\Omega^\mathrm{D}} \frac{1}{2} v \left(\left(w\mathbf{b} + (\phi\mathbf{b})^\mathrm{D} \right) \cdot n + \left(w - \phi^\mathrm{D} \right) \alpha^F \right) . \tag{2.31}$$

They based α^F on the average in the boundary element ($\mathbf{b}^\mathrm{avg}$) and the Dirichlet value $\mathbf{b}^\mathrm{D}$. For walls, defined by $\mathbf{b}^\mathrm{D} \cdot \mathbf{n} = 0$, the above term becomes

$$\int_{\partial\Omega^\mathrm{D}} v \left(\frac{1}{2} w\ \mathbf{b} \cdot \mathbf{n} + \left(w - \phi^\mathrm{D} \right) |\mathbf{b}^\mathrm{avg} \cdot \mathbf{n}| \right) . \tag{2.32}$$

They used this discretization for the nonlinear convective term, so that $\mathbf{b} = \mathbf{u}$ and $\phi = u_i$ for some direction i. Clearly this requires a Dirichlet value for all directions of the velocity, while the non-normal components of $\mathbf{u}$ are not physically relevant to the convection at a wall.

Many other authors have extended their numerical fluxes to the Dirichlet boundary in different but similar ways, with similar problems. Examples include [19, 33, 36] for the local Lax-Friedrichs flux, [37] for the Lesaint-Raviar numerical flux, and [38, 39] for Riemann-solved artificial compressibility flux.[2] These papers have treated Dirichlet faces like they are internal faces, except with the neighbor values replaced by the Dirichlet values. This is motivated by an understandable desire to impose the Dirichlet boundary condition weakly, but note that the present treatment is also weak, because we only use the moments $\int_{\partial\Omega^\mathrm{D}} v \min\left(0, \mathbf{n} \cdot \mathbf{b}^\mathrm{D}\right) \phi^\mathrm{D}$ for all test functions v, not the inflow value at every point on $\partial\Omega^\mathrm{D}$. The correct Eqs.

[2] though in that last case it is possible that the boundary treatment is at least consistent between the continuity and momentum equations

2.20–2.21 can also be found in some previous works, such as [13, pp. 27–28] and [40].

Perhaps the inaccuracies described above have little impact on the numerical solution in practical calculations, because most convected quantities are also diffused, which often weakly enforces the Dirichlet boundary condition for ϕ at $\partial\Omega^{\mathrm{D}}$. Superfluous convective boundary terms, such as Eqs. 2.30 and 2.31, would then effectively change the penalty parameter in the SIP method at Dirichlet boundaries.

2.6. Implementation

All flow simulations are performed with an in-house solver `DGFlows`. Its distinguishing feature is that it can be coupled with another in-house solver for particle transport, called `Phantom-`S_N. This has been used to solve the coupled flow and neutron transport equations for modeling a theoretical nuclear reactor that is based on a liquid fuel [41].

As mentioned in section 2.1.2, the basis functions are modal and hierarchical. As is standard in finite element implementations, they are defined on a 'local' reference element, which is mapped to the 'global', physical elements in the mesh. Sometimes this local-global mapping is affine, such as when the global elements are triangles, tetrahedrons, rectangles, or rectangular parallelepipeds. In that case the basis functions can be orthonormalized on the local element, which results in an orthogonal basis on the global elements, so that the mass matrix is diagonal. This can be a minor performance gain, especially if one were to use a numerical method that requires frequent projections of the numerical solution onto lower-order polynomial spaces, such as a p-multigrid solver, or a dynamic large eddy simulation.

All integrals are evaluated with a quadrature set that is sufficiently accurate to negate the polynomial aliasing effect that has plagued other DG solvers. (See, e.g., [42].) This is feasible because there are only $(\mathcal{P}+d)!/(d!\,\mathcal{P}!)$ degrees of freedom in a d-dimensional element with a polynomial order $\mathcal{P}$. In the limit of large $\mathcal{P}$ in three dimensions, this is $d! = 6$ times less than the $(\mathcal{P}+1)^d$ degrees of freedom in some nodal bases. The abscissa and the weights are taken from Solin *et al.* [43]. We store the values and derivatives of the basis functions on the quadrature set for a fast evaluation of integrals and numerical solutions. All results in this thesis remained unchanged when the accuracy of the quadrature was increased.

All meshes were generated with the open-source software tool `Gmsh` [44].

The linear systems are solved with the MPI-based software library `PETSc` [45, 46]. We use a conjugate gradient (CG) method for the pressure equation, and a GMRES method for the enthalpy and momentum equations, which are asymmetric.

The computations are parallelized by partitioning the mesh with the software package `METIS` [47]. Each core is assigned one partition. The parallel matrix preconditioner is a standard block Jacobi method. We use `PETSc`'s implementations of the pipelined Krylov methods, which require fewer inner products (meaning fewer global MPI reductions) at the cost of more serial computations (see [48] for pipelined CG, and [49] for pipelined GMRES). This provided better performance on multi-node computations.

The preconditioner for the submatrix within a process depends on the specific calculation. If the matrix remains constant for all time steps, then we use an incomplete LU (or Cholesky) decomposition with zero fill (i.e., ILU(0) or ICC(0)). These can be reused every time that the system is solved. If the matrix is different for each time step, then the incomplete decompositions are too expensive, and we instead use a block Gauss-Seidel method, where the degrees of freedom within an element are treated as one block.

The ordering of the elements within a core is random. It would likely be more efficient to renumber them based on the flow direction, so that a block Gauss-Seidel method for the transport equations would correspond to the upwind direction. Fidkowski *et al.* [50] and Diosady and Darmofal [51] have shown significantly increased performance by forming lines of maximum coupling between elements, and solving a block-tridiagonal system along each line.

Another obvious improvement is based on the hierarchical high-order solution space, which strongly suggests a p-multigrid method, based on Galerkin projection into lower-order polynomial spaces. Multigrid methods can greatly reduce the computational time for the pressure Poisson equation, which is often the most expensive part of an incompressible flow solver. This idea has been tested successfully for DG discretizations as early as 2005 [52, 53], and has since been applied to convection-advection problems by several other groups (e.g., [50, 54], [7, pp. 63–73]).

More recently, the high algorithmic intensity and minimal coupling between elements in a DG discretization have been exploited for efficient implementations on graphical processing units (GPUs). Various groups have demonstrated huge potential gains (e.g., [55–57]).

Unfortunately these ideas have not made it into generic numerical software packages, and we have made no attempt to implement them into `DGFLOWS`, which is purely a research code. Linear solvers are not studied in this thesis.

2.7. Test Case: A Heated Backward-facing Step

Backward-facing steps have a long history as benchmark cases for CFD methods. We simulate a two-dimensional case with an expansion ratio of 2, combined with scalar transport. Fig. 2.1 shows the domain. The inlet velocity (at $x = -L_0$) is given by $u_2 = 0$ and $u_1 = u_1^{(\text{in})} \coloneqq -6\bar{u}(y - S)(y - 2S)/S^2$, so that the average inlet velocity is $\int_{y=S}^{2S} u_1^{(\text{in})}/S = \bar{u}$. The temperature is T_0 at the inlet and T_1 along the bottom wall; the other walls are isolated.

The Reynolds number can be defined in many ways; here

$$\mathrm{Re} \coloneqq \frac{2S\bar{u}}{\nu} \,. \tag{2.33}$$

Some other literature uses the maximum velocity, or the height before the expansion. Eq. 2.33 is used because $2S$ is the hydraulic diameter at the inlet, making it consistent with the standard definition of the bulk Reynolds number for three-dimensional channel flow and pipe flow, which we will also see in Chapter 5.

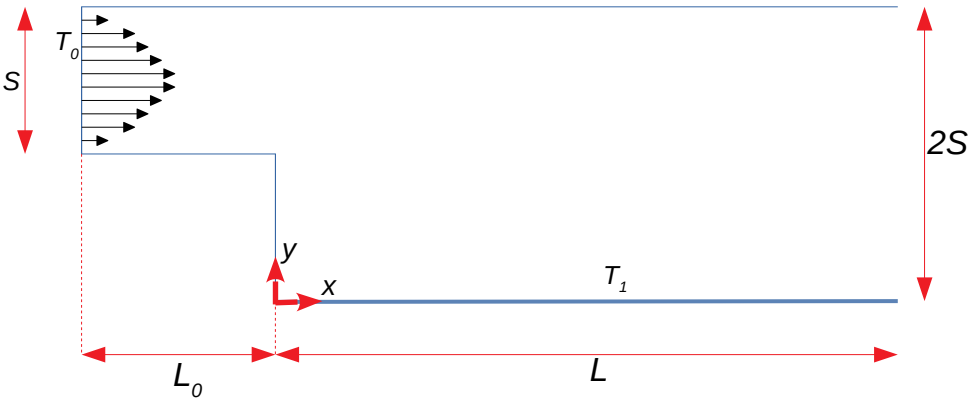


Figure 2.1: Geometry of the backward-facing step (not to scale).

The quantities of interest are the dimensionless velocity and temperature gradients along the bottom wall. Define the local Darcy friction factor

$$f_{\mathrm{D}} \coloneqq \frac{8\,\tau_{\mathrm{w}}^{(\mathrm{bottom})}}{\rho(\bar{u}/2)^2} = -\frac{8\,\nu}{(\bar{u}/2)^2}\,\mathbf{n}\cdot\nabla u_1\,, \tag{2.34}$$

where $\tau_{\mathrm{w}}^{(\mathrm{bottom})} = -\mu\,\mathbf{n}\cdot\nabla u_1$ is the wall shear stress along the bottom wall, and $(\bar{u}/2)$ is the bulk velocity after the expansion. The local Nusselt number is

$$\mathrm{Nu} \coloneqq \frac{S}{T_0 - T_1}\,\mathbf{n}\cdot\nabla T\,. \tag{2.35}$$

The domain should be long enough in order for the outlet not to influence the flow near the expansion. The present results were obtained with $L/S = 70$. The domain length before the expansion has little impact on the laminar solution; here we let $L_0 = S$. The results did not change when we ran the same simulation with $L/S = 55$.

Of course this does not mean that the flow is fully developed near the outlet. Far from the expansion, the velocity and temperature approach the analytical solutions

$$\begin{aligned} \lim_{x/S\to\infty} u_1 &= \frac{3}{2}\,\frac{\bar{u}}{2}\left(1 - \left(\frac{y-S}{S}\right)^2\right)\,, \\ \lim_{x/S\to\infty} T &= T_1\,, \end{aligned} \tag{2.36}$$

giving

$$\begin{aligned} \lim_{x/S\to\infty} f_{\mathrm{D}} &= 24\,\frac{\nu/S}{\bar{u}/2} = \frac{96}{\mathrm{Re}}\,, \\ \lim_{x/S\to\infty} \mathrm{Nu} &= 0\,, \end{aligned} \tag{2.37}$$

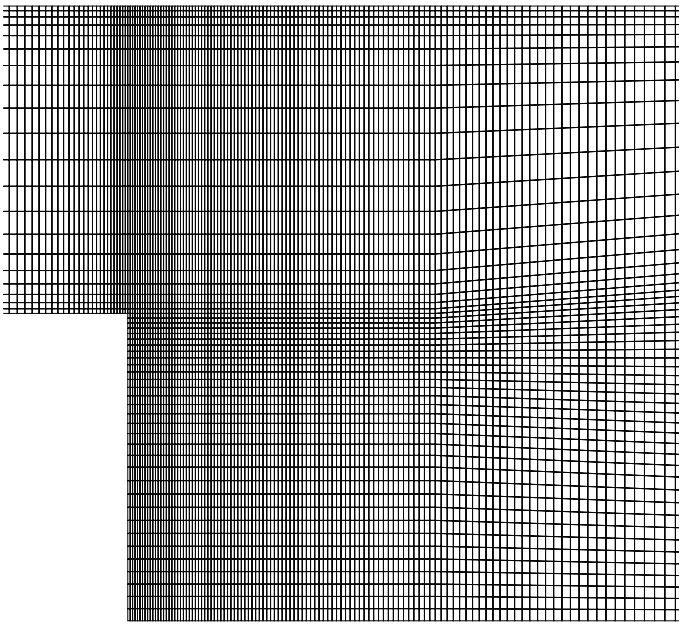


Figure 2.2: Computational mesh for the backward-facing step near the expansion edge.

(For comparison, the Darcy friction factor for laminar flow in a circular channel [i.e., Hagen-Poiseuille flow] is $f_\mathrm{D} = 64/\mathrm{Re}$.) The results will show that f_D and Nu are not close to these values near our outlet.

Fig. 2.2 shows the structured mesh. The mesh is refined near the expansion edge. There are approximately 35k elements, with a second-order polynomial approximation for the mass flux, pressure, and enthalpy, resulting in approximately 210k degrees of freedom per unknown.

An accurate solution depends on a sufficiently fine mesh near the expansion edge. The wall shear stress at the inlet is

$$\tau_\mathrm{w}^{(\mathrm{in})} = \mu \left.\frac{\partial u_1^{(\mathrm{in})}}{\partial y}\right|_{y=S} = 6\,\frac{\mu \bar{u}}{S} = 3\,\mathrm{Re}\frac{\mu^2}{\rho S^2}\,, \tag{2.38}$$

which can be used to define a wall shear velocity of $u_\tau \coloneqq \sqrt{\tau_\mathrm{w}^{(\mathrm{in})}/\rho} = (\nu/S)\sqrt{3\ \mathrm{Re}}$, and a dimensionless wall distance of $y^+ \coloneqq y u_\tau/\nu = (y/S)\sqrt{3\ \mathrm{Re}}$. The first element at the wall is placed at $y^+ = 1.0$.

Fig. 2.3 shows the results for $\mathrm{Re} = 1\,400$ and $\mathrm{Pr} = 0.7132$. These conditions were meant to reproduce one of the numerical test cases by Xie and Xi [58][3], who performed an unsteady simulation with the initial condition $u_1 = 0$ for $y < S$,

[3]The fact that $\mathrm{Pr} = 0.7132$ in their calculations was obtained from our private correspondence.

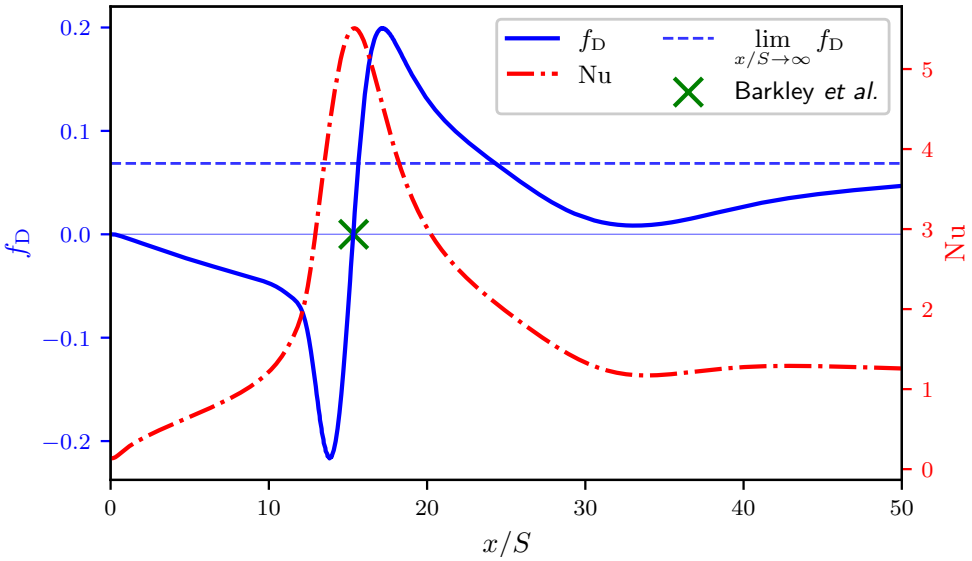


Figure 2.3: Steady-state local Darcy friction factor f_D (Eq. 2.34) and local Nusselt number Nu (Eq. 2.35) for the backward-facing step along the bottom wall. The cross (×) indicates the reattachment length (where $f_D = 0$) that was reported by Barkley *et al.* [61].

$u_1 = u_1^{(in)}$ for $y > S$, $u_2 = 0$, and $T = T_0$. They reported time-averaged results. We performed the same time-dependent calculation (with the pressure correction method that will be the topic of the next chapter), but we found that all transients die out, and we reached a steady state. Furthermore, our results differ substantially from Xie and Xi (not shown here). We nevertheless believe our results to be correct for two reasons. First, several other previous studies have also found a steady state in 2D at this Reynolds number (e.g., [59, 60]). Second, the location of our reattachment length (i.e., the largest value of x for which $f_D = 0$) is at $x/S = 15.342$, which agrees with the value of $x/S = 15.358$ that can be inferred by interpolating the data from Fig. 5 in Barkley *et al.* [61].

2.8. Discussion and Conclusion

The discrete convection in section 2.5 contains two corrections on previous literature that are essential to the consistency of the numerical method. First, the inflow at a Dirichlet boundary does not depend on an internal numerical value (ref. section 2.5.3). This contradicts a majority of papers on the DG method that we have seen. Second, the solution space of an advected quantity lies in the solution space of the pressure (ref. section 2.5.2). These points guarantee that the advection discretization is satisfied by the continuous advected quantity, even when the advecting field contains a numerical error. Phrased differently, if the Dirichlet boundary is not treated correctly, then the discretization of the full system of transport equations may be consistent, but the transport equations in isolation are not.

The numerical fluxes for the convection and diffusion discretizations are based on the well-known local Lax-Friedrichs and interior penalty methods, but we differ from the commonly accepted approach of averaging or maximizing the penalty parameters over a region, instead evaluating them in a pointwise manner. The merit of this is somewhat subjective, though we have argued that pointwise flux definitions are more in line with the local nature of the DG method. This is particularly true for the penalty term in the local Lax-Friedrichs flux, which is inconsistent for linear advection when α^F is averaged (ref. section 2.5.1, Eq. 2.24). Penalty terms can have a significant impact on stability, the solution quality, and the stiffness of the linear system. A numerical comparison between pointwise and averaged or maximized penalty parameters could therefore be interesting.

The penalty parameter in Eq. 2.16 could be an overestimate of the minimum value that achieves coercivity of the SIP method, because it was developed for a nodal basis. For the same minimum polynomial order, our modal basis functions have a substantially lower maximum polynomial order than the nodal basis functions (see section 2.1.2). We did not investigate alternative expressions for the penalty parameter.

The time-independent test case in section 2.7 shows good agreement with previous literature. It is also fairly simple. The following chapters will feature more challenging transient simulations, which double as further verification and validation of the spatial discretization that is described here.

References

[1] A. Hennink, M. Tiberga, and D. Lathouwers, *A Pressure-based solver for low-Mach number flow using a discontinuous Galerkin method,* Journal of Computational Physics **425**, 109877 (2021).

[2] B. Cockburn and C.-W. Shu, *Runge-Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems,* Journal of Scientific Computing **16**, 173 (2001).

[3] M. Dumbser, D. S. Balsara, E. F. Toro, and C.-D. Munz, *A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes,* Journal of Computational Physics **227**, 8209 (2008).

[4] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini, *Unified analysis of discontinuous Galerkin methods for elliptic problems,* SIAM Journal on Numerical Analysis **39**, 1749 (2002).

[5] K. Shahbazi, *An explicit expression for the penalty parameter of the interior penalty method,* Journal of Computational Physics **205**, 401 (2005).

[6] Y. Epshteyn and B. Rivière, *Estimation of penalty parameters for symmetric interior penalty Galerkin methods,* Journal of Computational and Applied Mathematics **206**, 843 (2007).

[7] K. Hillewaert, *Development of the discontinuous Galerkin method for high-resolution, large scale CFD and acoustics in industrial geometries*, Phd thesis, Université catholique de Louvain (2013).

[8] D. A. di Pietro and A. Ern, *Mathematical Aspects of Discontinuous Galerkin Methods* (Springer, 2012).

[9] B. Krank, M. Kronbichler, and W. A. Wall, *Wall modeling via function enrichment within a high-order DG method for RANS simulations of incompressible flow,* International Journal for Numerical Methods in Fluids **86**, 107 (2018).

[10] B. Krank, M. Kronbichler, and W. A. Wall, *A multiscale approach to hybrid RANS/LES wall modeling within a high-order discontinuous Galerkin scheme using function enrichment,* International Journal for Numerical Methods in Fluids **90**, 81 (2019).

[11] J. S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods* (Springer, New York, 2008).

[12] K. Shahbazi, P. F. Fischer, and C. R. Ethier, *A high-order discontinuous Galerkin method for the unsteady incompressible Navier-Stokes equations,* Journal of Computational Physics **222**, 391 (2007).

[13] G. Kanschat, *Discontinuous Galerkin Methods for Viscous Incompressible Flow*, 1st ed. (Teubner Research: Deutscher Universitäts-Verlag, Wiesbaden, 2007).

[14] V. Girault, B. Rivière, and M. F. Wheeler, *A Discontinuous Galerkin Method with Nonoverlapping Domain Decomposition for the Stokes and Navier-Stokes Problems,* Mathematics of Computation **74**, 53 (2005).

[15] V. John, *Finite Element Methods for Incompressible Flow Problems*, Springer Series in Computational Mathematics (Springer, Cham, 2016).

[16] B. Cockburn, G. Kanschat, and D. Schötzau, *An Equal-Order DG Method for the Incompressible Navier-Stokes Equations,* Journal of Scientific Computing **40**, 188 (2009).

[17] L. Botti and D. A. Di Pietro, *A pressure-correction scheme for convection-dominated incompressible flows with discontinuous velocity and continuous pressure,* Journal of Computational Physics **230**, 572 (2011).

[18] K. Shahbazi, *A Parallel High-Order Discontinuous Galerkin Solver For the Unsteady Incompressible Navier-Stokes Equations in Complex Geometries*, Phd thesis, University of Toronto (2007).

[19] B. Krank, N. Fehn, W. A. Wall, and M. Kronbichler, *A high-order semi-explicit discontinuous galerkin solver for 3d incompressible flow with application to dns and les of turbulent channel flow,* Journal of Computational Physics **348**, 634 (2017).

[20] D. A. Di Pietro, *Analysis of a discontinuous Galerkin approximation of the Stokes problem based on an artificial compressibility flux,* International Journal for Numerical Methods in Fluids **55**, 793 (2007).

[21] R. Hartmann, *Numerical Analysis of Higher Order Discontinuous Galerkin Finite Element Methods,* (2008).

[22] C. M. Klaij, J. J. W. van der Vegt, and H. van der Ven, *Space-time discontinuous Galerkin method for the compressible Navier-Stokes equations,* Journal of Computational Physics **217**, 589 (2006).

[23] B. Klein, B. Müller, F. Kummer, and M. Oberlack, *A high-order discontinuous Galerkin solver for low Mach number flows,* International Journal for Numerical Methods in Fluids **81**, 489 (2016).

[24] C. R. Ethier and D. a. Steinman, *Exact fully 3D Navier-Stokes Solutions for Benchmarking,* International Journal for Numerical Methods in Fluids **19**, 369 (1994).

[25] R. J. LeVeque, *Numerical Methods for Conservation Laws*, 2nd ed. (Birkhäuser, Basel; Boston; Berlin, 1992) p. 214.

[26] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge Texts in Applied Mathematics (Cambridge University Press, 2002).

[27] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*, 3rd ed. (Springer, Berlin, Heidelberg, 2009).

[28] M. E. Vázquez-Cendón, *Solving Hyperbolic Equations with Finite Volume Methods*, 1st ed. (Springer International Publishing, 2015) p. 188.

[29] A. R. Winters, R. C. Moura, G. Mengaldo, G. J. Gassner, S. Walch, J. Peiro, and S. J. Sherwin, *A comparative study on polynomial dealiasing and split form discontinuous Galerkin schemes for under-resolved turbulence computations,* Journal of Computational Physics **372**, 1 (2018), arXiv:1711.10180 .

[30] G. Vijayasundaram, *Transonic flow simulations using an upstream centered scheme of Godunov in finite elements,* Journal of Computational Physics **63**, 416 (1986).

[31] M. Piatkowski, S. Müthing, and P. Bastian, *A stable and high-order accurate discontinuous Galerkin based splitting method for the incompressible Navier-Stokes equations,* Journal of Computational Physics **356**, 220 (2018).

[32] B. Klein, F. Kummer, and M. Oberlack, *A SIMPLE based discontinuous Galerkin solver for steady incompressible flows,* Journal of Computational Physics **237** (2013), 10.1016/j.jcp.2012.11.051.

[33] M. de la Llave Plata, V. Couaillier, and M.-C. le Pape, *On the use of a high-order discontinuous Galerkin method for DNS and LES of wall-bounded turbulence,* Computers & Fluids **176**, 320 (2018).

[34] M. Tavelli and M. Dumbser, *A pressure-based semi-implicit space-time discontinuous Galerkin method on staggered unstructured meshes for the solution of the compressible Navier-Stokes equations at all Mach numbers,* Journal of Computational Physics **341**, 341 (2017).

[35] M. Tiberga, A. Hennink, J. L. Kloosterman, and D. Lathouwers, *A high-order discontinuous Galerkin solver for the incompressible RANS equations coupled to the k-ϵ turbulence model,* Computers & Fluids **212**, 104710 (2020).

[36] F. Zhang, J. Cheng, and T. Liu, *A reconstructed discontinuous Galerkin method for incompressible flows on arbitrary grids,* Journal of Computational Physics **418**, 109580 (2020).

[37] E. Ferrer and R. H. J. Willden, *A high order Discontinuous Galerkin Finite Element solver for the incompressible Navier-Stokes equations,* Computers & Fluids **46**, 224 (2011).

[38] F. Bassi, A. Crivellini, D. A. D. Pietro, and S. Rebay, *An artificial compressibility flux for the discontinuous Galerkin solution of the incompressible Navier-Stokes equations,* Journal of Computational Physics **218**, 794 (2006).

[39] M. Franciolini, A. Crivellini, and A. Nigro, *On the efficiency of a matrix-free linearly implicit time integration strategy for high-order Discontinuous Galerkin solutions of incompressible turbulent flows,* Computers & Fluids **159**, 276 (2017).

[40] N. Emamy, F. Kummer, M. Mrosek, M. Karcher, and M. Oberlack, *Implicit-explicit and explicit projection schemes for the unsteady incompressible Navier-Stokes equations using a high-order dG method,* Computers & Fluids **154**, 285 (2017).

[41] M. Tiberga, *Development of a high-fidelity multi-physics simulation tool for liquid-fuel fast nuclear reactors*, Phd thesis, Delft University of Technology (2020).

[42] G. Mengaldo, D. De Grazia, D. Moxey, P. Vincent, and S. Sherwin, *Dealiasing techniques for high-order spectral element methods on regular and irregular grids,* Journal of Computational Physics **299**, 56 (2015).

[43] P. Solin, K. Segeth, and I. Dolezel, *Higher-Order Finite Element Methods*, 1st ed. (Chapman and Hall/CRC, New York, 2003) p. 408.

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

[45] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini,

H. Zhang, and H. Zhang, *PETSc Users Manual*, Tech. Rep. ANL-95/11 - Revision 3.9 (Argonne National Laboratory, 2018).

[46] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, *Efficient management of parallelism in object oriented numerical software libraries,* in *Modern Software Tools in Scientific Computing*, edited by E. Arge, A. M. Bruaset, and H. P. Langtangen (Birkhäuser Press, 1997) pp. 163–202.

[47] G. Karypis and V. Kumar, *A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs,* SIAM Journal on Scientific Computing **20**, 359 (1998).

[48] P. Ghysels and W. Vanroose, *Hiding global synchronization latency in the preconditioned Conjugate Gradient algorithm,* Parallel Computing **40**, 224 (2014).

[49] P. Ghysels, T. J. Ashby, K. Meerbergen, and W. Vanroose, *Hiding Global Communication Latency in the GMRES Algorithm on Massively Parallel Machines,* SIAM Journal on Scientific Computing **35**, C48 (2013).

[50] K. J. Fidkowski, T. A. Oliver, J. Lu, and D. L. Darmofal, *p-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations,* Journal of Computational Physics **207**, 92 (2005).

[51] L. Diosady and D. Darmofal, *Discontinuous Galerkin Solutions of the Navier-Stokes Equations Using Linear Multigrid Preconditioning,* in *18th AIAA Computational Fluid Dynamics Conference* (Miami, FL, U.S.A., 2007).

[52] B. Helenbrook and H. Atkins, *Application of P-Multigrid to Discontinuous Galerkin Formulations of the Poisson Equation,* in *17th AIAA Computational Fluid Dynamics Conference* (Toronto, Ontario, 2005).

[53] H. L. Atkins and B. T. Helenbrook, *Numerical Evaluation of P-Multigrid Method for the Solution of Discontinuous Galerkin Discretizations of Diffusive Equations,* in *17th AIAA Computational Fluid Dynamics Conference* (Toronto, Ontario, 2005).

[54] H. Luo, J. D. Baum, and R. Löhner, *Fast p-Multigrid Discontinuous Galerkin Method for Compressible Flows at All Speeds,* AIAA Journal **46**, 635 (2008).

[55] Y. Xia, J. Lou, H. Luo, J. Edwards, and F. Mueller, *OpenACC acceleration of an unstructured CFD solver based on a reconstructed discontinuous Galerkin method for compressible flows,* International Journal for Numerical Methods in Fluids **78**, 123 (2015).

[56] A. Karakus, N. Chalmers, K. Świrydowicz, and T. Warburton, *A GPU accelerated discontinuous Galerkin incompressible flow solver,* Journal of Computational Physics **390**, 380 (2019).

[57] A. C. Kirby and D. J. Mavriplis, *GPU-Accelerated Discontinuous Galerkin Methods: 30x Speedup on 345 Billion Unknowns,* Journal of Computational Physics (2020), manuscript submitted for publication.

[58] W. A. Xie and G. N. Xi, *Fluid flow and heat transfer characteristics of separation and reattachment flow over a backward-facing step,* International Journal of Refrigeration **74**, 177 (2017).

[59] A. Fortin, M. Jardak, J. J. Gervais, and R. Pierre, *Localization of Hopf bifurcations in fluid flow problems,* International Journal for Numerical Methods in Fluids **24**, 1185 (1997).

[60] L. Kaiktsis, G. Em Karniadakis, and S. A. Orszag, *Unsteadiness and convective instabilities in two-dimensional flow over a backward-facing step,* Journal of Fluid Mechanics **321**, 157 (1996).

[61] D. Barkley, M. G. M. Gomes, and R. D. Henderson, *Three-dimensional instability in flow over a backward-facing step,* Journal of Fluid Mechanics **473**, 167 (2002).

3

Pressure Correction

3.1. Introduction

This chapter treats the temporal discretization of the transport equations 1.1a–1.1c with a constant density. The test cases at the end of this chapter use the discontinuous Galerkin method that was described in chapter 2, which provides additional confidence in the spatial discretization. This chapter can nevertheless be read independently.

We assume that the boundary of Ω does not depend on t, so that the spatial and temporal discretizations can be entirely uncoupled. The more general moving boundary problem has previously been approached with a space-time discontinuous Galerkin method [2–4]. Our independent variables lie in the space-time cylinder $(t, \mathbf{r}) \in (0, T) \times \Omega$, which suggests time stepping with a simple finite difference method.

3.1.1. Fully Discrete Linear System

The temporal discretizations for the momentum and the enthalpy are based on standard backward-difference formulae (BDF). For the mass flux this is straightforward: for a constant time step size δt,

$$\frac{\partial \mathbf{m}}{\partial t} \approx \frac{\gamma_0}{\delta t}\mathbf{m}^n + \sum_{i=1}^{q} \frac{\gamma_i}{\delta t}\mathbf{m}^{n-i} \,, \tag{3.1}$$

where $\mathbf{m}^n$ is the mass flux at time step n. The weights $\{\gamma_i\}_{i=0}^{q}$ are listed in Table 3.1.

If the density depends on the temperature, then the temporal discretization of the enthalpy equation becomes more involved, as will be explained in great detail in

Parts of this chapter have been published in [1].

Table 3.1: Coefficients for the backward difference formula of various orders.

	γ_0	γ_1	γ_2	γ_3
BDF1	1	-1		
BDF2	3/2	-2	1/2	
BDF3	11/6	-3	3/2	-1/3

chapter 4. In this chapter ρ is simply constant, and so the finite difference scheme for $\partial(\rho h)/\partial t$ is straightforward.

The fully discrete transport equations can be written as

$$-\boldsymbol{D}\underline{\boldsymbol{m}}^n + \boldsymbol{C}\underline{\boldsymbol{p}}^n = -\underline{\boldsymbol{r}} \tag{3.2a}$$

$$\frac{\gamma_0}{\delta t}\boldsymbol{M}\,\underline{\boldsymbol{m}}^n = -\boldsymbol{N}\underline{\boldsymbol{m}}^n - \boldsymbol{D}^{\mathsf{T}}\underline{\boldsymbol{p}}^n + \underline{\boldsymbol{f}} \tag{3.2b}$$

$$\frac{\gamma_0}{\delta t}\boldsymbol{T}\,\underline{\boldsymbol{h}}^n = -\boldsymbol{F}\underline{\boldsymbol{h}}^n + \underline{\boldsymbol{q}}\,, \tag{3.2c}$$

where $\underline{\boldsymbol{p}}$, $\underline{\boldsymbol{m}}$, and $\underline{\boldsymbol{h}}$ are the solution vectors, containing the coefficients of the basis functions, and

- $\boldsymbol{M}$ is the mass matrix (i.e., the Gram matrix of the basis functions);
- $\boldsymbol{D}$ corresponds to the divergence operator in Eq. 2.5;
- $\boldsymbol{C}$ corresponds to the pressure stabilization in Eq. 2.8, which is zero for mixed-order discretizations;
- $\boldsymbol{N}$ contains the implicit terms of the convection and diffusion discretizations of the momentum equation;
- $\boldsymbol{F}$ contains the implicit terms of the convection and diffusion discretizations of the enthalpy equation;
- $(\gamma_0/\delta t)\boldsymbol{T}$ corresponds to the implicit part of the BDF scheme for the enthalpy, that is, the coefficient of h^n in

$$\frac{\partial(\rho h)}{\partial t} \approx \sum_{i=0}^{q} \frac{\gamma_i}{\delta t}(\rho h)^{n-i}\,. \tag{3.3}$$

  For the constant-density flow in this chapter, $\boldsymbol{T} = \rho\boldsymbol{M}$. For the variable-density flow in the next chapter, $\boldsymbol{T}$ depends on how $(\rho h)^n$ is approximated (see section 4.3);
- $\underline{\boldsymbol{r}}$ corresponds to the rhs of the discrete continuity equation (Eq. 2.7);
- $\underline{\boldsymbol{f}}$, and $\underline{\boldsymbol{q}}$ collect various explicit terms, including those from the temporal discretization, and from the boundary conditions.

The matrices $\boldsymbol{M}$ and $\boldsymbol{D}$ and the vector $\underline{\boldsymbol{r}}$ do not depend on the unknowns $\left(\underline{\boldsymbol{p}}, \underline{\boldsymbol{m}}, \underline{\boldsymbol{h}}\right)$, but the other terms depend on the fluid properties, and therefore on $\underline{\boldsymbol{h}}$. The most important nonlinearity is due to $\boldsymbol{N}$, which depends on the convective field $(1/\rho)m$.

The pressure-based linear system 3.2a–3.2b for the discrete continuity and momentum equations is very stiff, because the equations form a saddle point problem. Therefore these equations are approximated with a time-splitting method, as explained in the following section.

We use a second-order BDF2 scheme ($q = 2$ in Eqs. 3.1 and 3.3), thereby following previous DG literature (e.g., [5–7]). Discontinuous Galerkin methods for the Navier-Stokes equations have traditionally been associated with high-order temporal accuracy, which can easily be achieved in density-based formulations that solve a coupled system of transport equations. However, splitting methods have an inherent error of order $3/2$ in the H^1-norm [8], so that high-order BDF schemes have no merit.

3.2. Pressure Correction Method

The pressure correction method is used to split the continuity and the momentum equations, so that they can be solved in a segregated way, which is much cheaper than a coupled solver. This technique has been thoroughly analyzed in the context of many spatial discretizations. See, for example, Saleri and Veneziani [9] for analyses based on LU-decompositions of system of transport equations, and the review by Guermond *et al.* [8] for a comparison with the alternatives to pressure correction, namely velocity correction and consistent splitting methods.

The pressure correction method for an equal-order discontinuous Galerkin discretization differs from the spatial discretizations in the above references, because of the pressure stabilization term in the continuity equation 3.2a. This couples the discrete continuity and momentum equations more tightly. This section summarizes the time-splitting scheme with pressure stabilization.

The momentum equation is solved with a known vector $\hat{\underline{\boldsymbol{p}}}$ instead of the unknown pressure $\underline{\boldsymbol{p}}^n$ to obtain a predictor $\hat{\underline{\boldsymbol{m}}}$ for the mass flux:

$$\left(\frac{\gamma_0}{\delta t}\boldsymbol{M} + \boldsymbol{N}\right)\hat{\underline{\boldsymbol{m}}} = -\boldsymbol{D}^{\mathsf{T}}\hat{\underline{\boldsymbol{p}}} + \underline{\boldsymbol{f}} \,. \tag{3.4}$$

These are then corrected to find the solutions at the new time step:

$$\underline{\boldsymbol{p}}^n = \hat{\underline{\boldsymbol{p}}} + \delta\underline{\boldsymbol{p}} \,, \tag{3.5}$$

$$\underline{\boldsymbol{m}}^n = \hat{\underline{\boldsymbol{m}}} + \delta\underline{\boldsymbol{m}} \,. \tag{3.6}$$

Subtracting Eq. 3.4 from Eq. 3.2b gives $((\gamma_0/\delta t)\boldsymbol{M} + \boldsymbol{N})\,\delta\underline{\boldsymbol{m}} = -\boldsymbol{D}^{\mathsf{T}}\delta\underline{\boldsymbol{p}}$. The idea of the pressure correction method is that $\boldsymbol{N}\,\delta\underline{\boldsymbol{m}}$ is of a higher order in δt than the other terms, and therefore it can be neglected, giving

$$\delta\underline{\boldsymbol{m}} = -\frac{\delta t}{\gamma_0}\boldsymbol{M}^{-1}\boldsymbol{D}^{\mathsf{T}}\delta\underline{\boldsymbol{p}} \,. \tag{3.7}$$

This approximation means that the momentum equation 3.2b is not exactly satisfied by the solution pair $(\underline{\boldsymbol{p}}^n, \underline{\boldsymbol{m}}^n)$, though it can be made to satisfy the continuity equation 3.2a exactly. Left-multiply Eq. 3.7 by $\boldsymbol{D}$ and use Eq. 3.2a to eliminate $\boldsymbol{D}\underline{\boldsymbol{m}}^n$, to find

$$\boldsymbol{C}\underline{\boldsymbol{p}}^n + \underline{\boldsymbol{r}} - \boldsymbol{D}\underline{\hat{\boldsymbol{m}}} = -\frac{\delta t}{\gamma_0}\boldsymbol{D}\boldsymbol{M}^{-1}\boldsymbol{D}^\intercal\delta\underline{\boldsymbol{p}}\ , \tag{3.8}$$

which can be rearranged to

$$\left(\boldsymbol{A}^{\mathrm{LDG}} + \frac{\gamma_0}{\delta t}\boldsymbol{C}\right)\delta\underline{\boldsymbol{p}} = \frac{\gamma_0}{\delta t}\left(-\underline{\boldsymbol{r}} + \boldsymbol{D}\underline{\hat{\boldsymbol{m}}} - \boldsymbol{C}\underline{\hat{\boldsymbol{p}}}\right) \tag{3.9}$$

with

$$\boldsymbol{A}^{\mathrm{LDG}} \coloneqq \boldsymbol{D}\boldsymbol{M}^{-1}\boldsymbol{D}^\intercal\ . \tag{3.10}$$

In the seminal paper on this method, Chorin [10] simply had $\underline{\hat{\boldsymbol{p}}} = \underline{0}$, but it was shown later that the incremental approach with $\underline{\hat{\boldsymbol{p}}} = \underline{\boldsymbol{p}}^{n-1}$ is more accurate [11], and that is what we use. It may be tempting to use a higher-order approximation of $\underline{\boldsymbol{p}}^n$ for $\underline{\hat{\boldsymbol{p}}}$, but this is only conditionally stable [8]. If the convection and the diffusion of the momentum are treated in a time-explicit manner, then $\boldsymbol{N} = 0$, and the pressure correction method yields an exact solution to Eqs. 3.2a–3.2b, regardless of the choice of $\underline{\hat{\boldsymbol{p}}}$.

Splitting the momentum and continuity equations has created the need for artificial initial and boundary conditions for the pressure, which are not present in the original system of equations (at least not on Dirichlet boundaries). The initial condition is not a practical problem; the pressure can be inferred from the initial condition for the mass flux, or the pressure correction method can be iterated within the first time step. Often the initial transient behavior is not physically relevant, in which case one can also just set $\underline{\boldsymbol{p}}^0 = \underline{0}$.

The pressure boundary condition would have been explicit if the pressure correction method had been derived at the differential level from the continuous equations 1.1a–1.1b. This would have resulted in a continuous Poisson equation ($\nabla^2 p = \ldots$) that needs to be supplemented with boundary conditions before it can be discretized. This could be seen as an argument in favor of 'algebraic' splitting methods, which are derived from the coupled discrete linear system 3.2a–3.2b, as was done above. Nevertheless, even the discrete Poisson-like pressure equation 3.9 contains implicit artificial boundary conditions, which are inherent to the construction of the discrete divergence operator $\boldsymbol{D}$ in Eq. 2.5.

In fact, Shahbazi *et al.* [5] have pointed out that $\boldsymbol{A}^{\mathrm{LDG}}$ (Eq. 3.10) is effectively a local discontinuous Galerkin (LDG) discretization for a diffusion operator with homogeneous Neumann boundary conditions at the walls and the inlet, so it can be replaced by an SIP diffusion operator

$$\boldsymbol{A}^{\mathrm{LDG}} \approx \boldsymbol{A}^{\mathrm{SIP}}\ , \tag{3.11}$$

which has a smaller stencil (as was discussed in section 2.1.1). Note that $\boldsymbol{A}^{\mathrm{SIP}}$ is equipped with explicit boundary conditions, and the time-splitting scheme is no

longer of the algebraic kind. A consequence of using this SIP pressure matrix is that the continuity equation 3.2a does not hold exactly. In the test cases that follow, we have not noticed a difference between using LDG and SIP pressure discretizations, which is in line with previous findings (e.g., [12, pp. 33–45]).

It has long been known that the artificial boundary condition is one of the main drawbacks of time-splitting methods [13]. It creates a numerical boundary layer that prevents the pressure from converging with second-order temporal accuracy in the L^2-norm. The velocity is second-order accurate in time in the L^2-norm, but not in the H^1-norm [14].

This problem of a numerical boundary layer can be remedied with the so-called rotational pressure correction method that was introduced by Timmermans *et al.* [15], and popularized by a rigorous error analysis due to Guermond and Shen [14]. They showed that the rotational correction results in a consistent pressure boundary condition, and that this improves the orders of convergence for the pressure in the L^2-norm, and the velocity in the H^1-norm, with the exact orders depending on the geometry and the type of boundary conditions. See also the extensive numerical tests and heuristic explanation in [16]. Piatkowski *et al.* [17] have combined rotational pressure correction with a discontinuous Galerkin method for the spatial discretization.

While these works have demonstrated a substantial improvement of the temporal error, they are based on the assumption of a homogeneous density and viscosity. Deteix and Yakoubi [18] have shown how to incorporate a variable viscosity, at the expense of having to solve two poisson equations for every pressure correction step. We have not extended the rotational pressure correction method to our DG solver with a variable viscosity.

In summary, our algorithm to find the solution vectors $\underline{\boldsymbol{p}}^n$, $\underline{\boldsymbol{m}}^n$, $\underline{\boldsymbol{h}}^n$ at a new time step n is as follows.

1. Obtain predictors for $(k/c_p)^*$, and $\mathbf{m}^*$ with a second-order extrapolation from previous time steps:
$$(\cdot)^* = 2(\cdot)^{n-1} - (\cdot)^{n-2} \,. \tag{3.12}$$

2. Solve for the enthalpy $\underline{\boldsymbol{h}}^n$ at the new time step, using the above predictors for the diffusion constant ($(k/c_p)^*$) and the advecting field ($\mathbf{m}^*$).

   If the density depends on the temperature, as in chapter 4, then the implicit time term is also approximated with a predictor $\rho^* = 2\rho^{n-1} - \rho^{n-2}$, using either of the methods that are explained in Section 4.3.

3. Solve Eq. 3.4 for $\underline{\hat{\boldsymbol{m}}}$. The matrix $\boldsymbol{N}$ depends on the fluid properties, which are evaluated at the new time step as a function of h^n. The convective field is estimated as $(1/\rho^n)\mathbf{m}^*$.

4. Solve the pressure Poisson equation 3.9 for $\delta\underline{\boldsymbol{p}}$, possibly replacing the LDG discretization by an SIP method (Eq. 3.11), and correct the pressure and the mass flux with Eqs. 3.5–3.7.

The manufactured solutions in section 3.3 will show full second-order temporal accuracy in the enthalpy and the mass flux, even if the fluid properties are non-trivial functions of the enthalpy.

3.3. Verification with Manufactured Solutions

The numerical method and its implementation are verified with two manufactured solutions: the well-known Taylor-Green vortex with constant fluid properties, and a variable-property manufactured solution with a wall and an outflow boundary.

Our experience with `DGFlows` has been that the convergence results depend critically on a careful calculation of the error, which is defined in the L^2-norm as

$$\frac{\|\phi - \phi^{\text{ex}}\|_2}{\|\phi^{\text{ex}}\|_2} = \sqrt{\frac{\sum_{T\in\mathcal{T}} \int_T (\phi - \phi^{\text{ex}})^2}{\int_\Omega (\phi^{\text{ex}})^2}} \tag{3.13}$$

for a quantity ϕ with an exact solution ϕ^{ex}. Each integral in the numerator is evaluated with a numerical quadrature, resulting in a large sum over the squares of small numbers. A naive implementation gives very large rounding errors. We therefore perform the double summation over the elements and the quadrature points with the Kahan summation algorithm [19] and a 128-bit floating point number.

All integrals in the weak forms are evaluated with a quadrature set with the usual polynomial accuracy of $(3\mathcal{P}_m - 1)$, and we verified that this is sufficient to integrate up to machine precision by comparing the results with a higher-order quadrature set of polynomial accuracy $(3\mathcal{P}_m + 10)$. This is not surprising, since the Taylor-Green vortex solution in section 3.3.1 is smooth, whereas the exact solution in section 3.3.2 is a polynomial, and so are the corresponding forcing terms.

3.3.1. Taylor-Green Vortex

The first manufactured solution is the Taylor-Green Vortex, which is incompressible and has constant fluid properties. We include a passive scalar temperature field with this well-known analytical solution. The enthalpy is $h = c_p T$. The exact solution is

$$\begin{aligned}
\mathbf{u}^{\text{ex}} &= \exp(-2\tilde{t}) \begin{bmatrix} -\cos(\tilde{x})\sin(\tilde{y}) \\ +\sin(\tilde{x})\cos(\tilde{y}) \end{bmatrix}, \\
p^{\text{ex}} &= -\frac{\rho}{4}\exp(-4\tilde{t})\left(\cos(2\tilde{x}) + \cos(2\tilde{y})\right), \\
T^{\text{ex}} &= \exp(-2\tilde{t}/\text{Pr})\cos(\tilde{x})\cos(\tilde{y}),
\end{aligned} \tag{3.14}$$

on a domain $x, y \in [-L, L]$ with Dirichlet boundary conditions and $0 < t \leq 1$, where

$$\tilde{t} \coloneqq \frac{\nu t}{(L/(n\pi))^2}, \quad \tilde{x} \coloneqq \frac{x}{L/(n\pi)}, \quad \tilde{y} \coloneqq \frac{y}{L/(n\pi)}, \tag{3.15}$$

and n must be a positive integer in order for $\int_\Omega p = 0$. This solves the transport equations with $\mathbf{F} = [0, 0]$ and $Q = 0$.

Fig. 3.1 shows the temporal convergence for $L = 1$, $n = 1$, $\mu = 0.01$, $\rho = 1$, $\mathrm{Pr} = 100$, and a fourth-order polynomial space for the mass flux (i.e., $\mathcal{P}_m = 4$). We performed the same numerical experiments by independently varying the Prandtl number (to $\mathrm{Pr} = 1$), and the polynomial order (to $\mathcal{P}_m = 2$), all of which yielded similar results. All errors saturate at small time steps, where the spatial discretization error dominates.

Table 3.2 shows the spatial convergence at the smallest time step that was tested. It also includes equal-order and mixed-order results for a polynomial orders. All quantities of polynomial order $\mathcal{P}$ appear to converge as $\mathcal{O}\left(\ell^{\mathcal{P}+1}\right)$, where $\ell \propto 1/N_y$ is the characteristic mesh length, except for the equal-order cases, where the order of convergence for the pressure is in the range $[\mathcal{P} + 1/2, \mathcal{P} + 1]$. The spatial convergence rates for the high-order polynomial cases is harder to make out from the available data, because the temporal error is still significant, as could also be seen in Fig 3.1.

The Taylor-Green vortex is of course a strange test case, in that it does not feature any scale separation: its Fourier transform is comprised of Dirac delta functions. The solution is an eigenfunction of the diffusive terms, so there is no interaction between the transport terms. Furthermore, it has Dirichlet boundary conditions where there is outflow, so that the continuity equation is over-constrained. This manifests itself in a stiff linear system for the pressure, though its ubiquity in the literature suggests that the Taylor-Green vortex is very easy to simulate. The next section features a more challenging manufactured solution.

3.3.2. Variable-property Manufactured Solution

This section features a manufactured solution with temperature-dependent transport properties, so that the momentum and enthalpy transport equations are coupled. It has walls and an outflow boundary condition; contrary to the Taylor-Green vortex, it is well-posed. The domain is $(0, L)\times(-1, 1)$; see Fig. 3.2. We let $L = 10$ in all calculations, and use square elements, so that there are $N_x = (L/2)N_y$ elements in the x-direction. The inflow boundary at $x = 0$ has Dirichlet boundary conditions. The goal of the manufactured solutions is obviously not to model a particular physical phenomenon, but our configuration is vaguely reminiscent of a pipe flow with walls at $y = \pm 1$ that is heated asymmetrically, resulting in skewed velocity profiles.

We use a Neumann boundary condition for the temperature at the outlet, because this is the most common choice in practical applications. The imposed heat flux (q^{N}) follows from the known exact solution. We also tried imposing a Dirichlet boundary condition for the temperature, and found that it makes a negligible difference in the numerical errors.

We choose the polynomial manufactured solution

$$\begin{aligned} \mathbf{m}^{\mathrm{ex}} &= (1 + t^3)\frac{1}{L^3}\begin{bmatrix} y\left(2x^3/3 - Lx^2\right) \\ (y-1)(y+1)x(L-x) \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \\ p^{\mathrm{ex}} &= (1 + t^3)\ (L - x)^3\ , \\ T^{\mathrm{ex}} &= (1 + t^3)\ (2 - y)((x - L)/L)^2/6 \end{aligned} \tag{3.16}$$

with $0 < t \leq 1$, which satisfies $m_2 = 0$ on $y = \pm 1$, and $\nabla \cdot \mathbf{m} = 0$. The addition

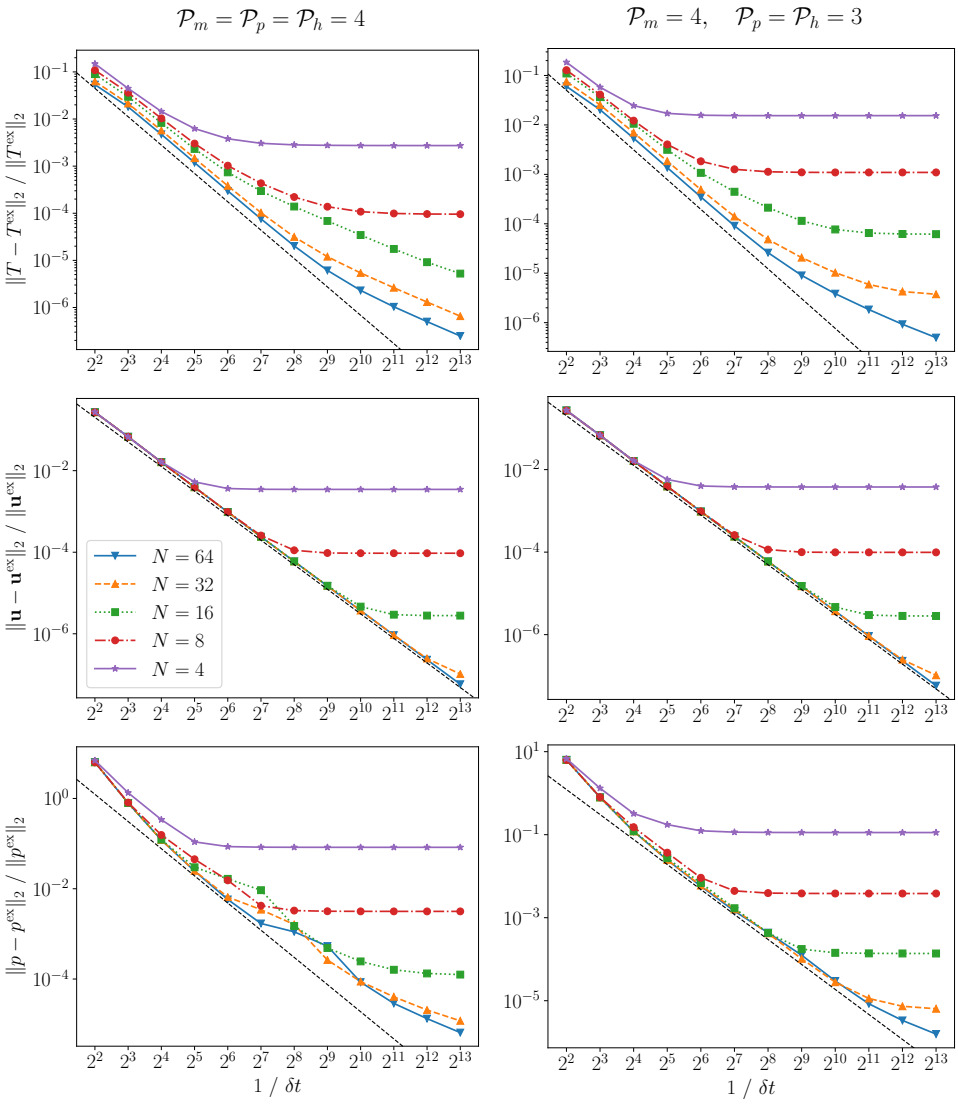


Figure 3.1: Convergence toward the 2D Taylor vortex (Eq. 3.14) at time $\tilde{t} = 1$ with temporal refinement for meshes with N^2 square elements. The black dashed lines indicate ideal second-order convergence in δt.

Table 3.2: Convergence toward the Taylor-Green vortex in Eq. 3.14 with spatial refinement, keeping δt fixed. The data in the last two blocks are taken from the highest temporal refinement in Fig. 3.1.

	temperature		velocity		pressure	
N_y	error	conv	error	conv	error	conv
Equal order $(\mathcal{P}_m = \mathcal{P}_h = \mathcal{P}_p = 1)$, $\delta t = 2^{-12}$:						
2^2	2.93e-1		1.13e0		1.17e0	
2^3	1.02e-1	1.53	5.25e-1	1.10	1.34e0	-0.19
2^4	2.73e-2	1.90	1.59e-1	1.73	5.84e-1	1.20
2^5	6.72e-3	2.02	4.14e-2	1.94	1.80e-1	1.69
2^6	1.66e-3	2.02	1.04e-2	1.99	5.34e-2	1.76
2^7	4.11e-4	2.01	2.60e-3	2.00	1.69e-2	1.66
2^8	1.02e-4	2.01	6.50e-4	2.00	6.05e-3	1.48
Mixed order $(\mathcal{P}_m = 2,\ \mathcal{P}_h = \mathcal{P}_p = 1)$, $\delta t = 2^{-12}$:						
2^2	2.70e-1		2.57e-1		2.20e0	
2^3	7.25e-2	1.90	2.13e-2	3.59	4.01e-1	2.46
2^4	1.70e-2	2.09	1.62e-3	3.72	5.43e-2	2.88
2^5	4.14e-3	2.03	1.49e-4	3.44	8.67e-3	2.65
2^6	1.04e-3	2.00	1.64e-5	3.18	1.71e-3	2.34
2^7	2.61e-4	1.99	1.97e-6	3.06	3.86e-4	2.15
2^8	6.59e-5	1.99	3.25e-7	2.60	9.28e-5	2.06
Mixed order $(\mathcal{P}_m = 4,\ \mathcal{P}_h = \mathcal{P}_p = 3)$, $\delta t = 2^{-13}$:						
2^2	1.54e-2		3.79e-3		1.13e-1	
2^3	1.09e-3	3.82	9.87e-5	5.26	3.82e-3	4.88
2^4	6.17e-5	4.15	2.83e-6	5.12	1.37e-4	4.80
2^5	3.75e-6	4.04	1.04e-7	4.77	6.40e-6	4.42
2^6	4.98e-7	2.91	5.80e-8	0.84	1.57e-6	2.02
Equal order $(\mathcal{P}_m = \mathcal{P}_p = \mathcal{P}_h = 4)$, $\delta t = 2^{-13}$:						
2^2	2.73e-3		3.47e-3		8.27e-2	
2^3	9.56e-5	4.84	9.46e-5	5.20	3.15e-3	4.71
2^4	5.24e-6	4.19	2.79e-6	5.08	1.25e-4	4.65
2^5	6.55e-7	3.00	1.03e-7	4.76	1.19e-5	3.40
2^6	2.49e-7	1.40	5.79e-8	0.83	6.51e-6	0.86

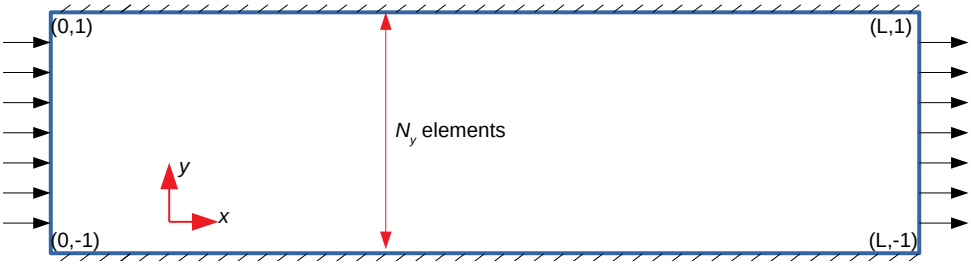


Figure 3.2: Domain of the manufactured solutions in Sections 3.3.2 and 4.5.1.

of the constant $[2, 0]$ to $\mathbf{m}^{\mathrm{ex}}$ ensures that $m_1 > 0$ everywhere, so that there is no backflow at the outlet, and no outflow at any of the Dirichlet boundary conditions. The density is $\rho = 1$, but the transport properties are non-trivial: $\mu = 0.1 + T(1-T)$ and $k = \mu c_p/\mathrm{Pr}$, with $\mathrm{Pr} = 1$. The solution is depicted in Fig. 3.3.

Fig. 3.4 displays the temporal convergence. We consider various meshes, varying the number of elements and the spatial polynomial orders. The equal-order case does not appear to suffer from the inf-sup instability for small δt. The velocity and the temperature converge with second-order accuracy in δt until the error saturates when the spatial error starts to dominate the temporal error. The order of convergence for the pressure is slightly lower, in the range [1.5,2.0]. These orders of convergence for the velocity and pressure agree with what is found in previous literature on constant-property incompressible flows (e.g., [5], [20]).

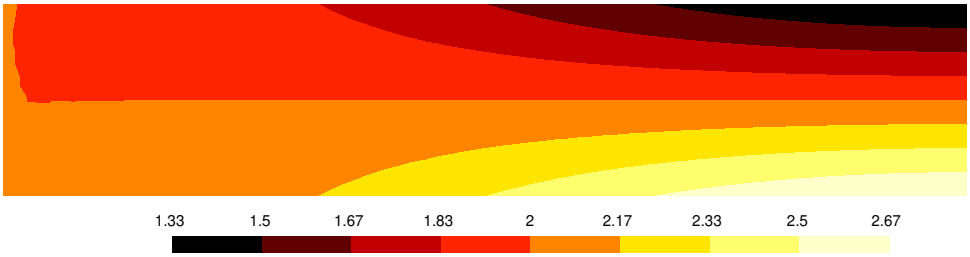
The spatial rates of convergence are in Table 3.3. As the mesh is refined, the mixed-order discretization displays $\mathcal{O}\left(\ell^{\mathcal{P}+1}\right)$ convergence for all quantities of polynomial order $\mathcal{P}$, though the convergence rate of the velocity saturates at high spatial refinement, as the temporal error starts to become significant. The equal-order discretization has the same convergence rates, meaning that the velocity shows hyperconvergence, with the error in u behaving as $\mathcal{O}\left(\ell^{\mathcal{P}_u+2}\right)$.

3.4. Validation with Flow Past a Circular Obstacle

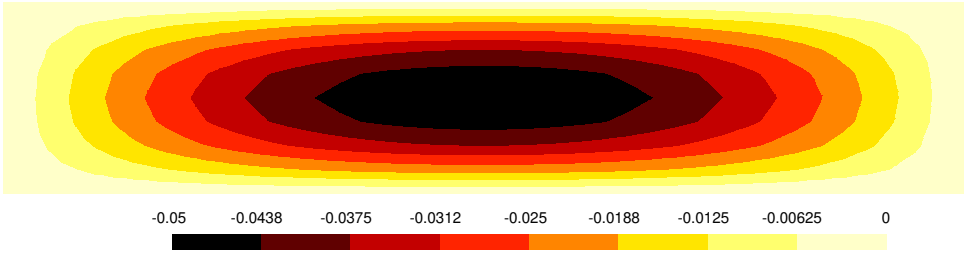
We computed laminar flow past a circular cylinder to validate our numerical method. This features a Von Kármán vortex street in the wake of the obstacle. The results of this well-known benchmark case can be compared to experiments and to other direct numerical simulations.

Fig. 3.5 shows the computational domain. The velocity is fixed at $[u_\infty, 0]$ on the left, top, and bottom parts of the domain. A subscript ∞ denotes a far-field value. The right side (at $x = L$) is an outlet with homogeneous Neumann boundary conditions (i.e., $q^{\mathrm{N}} = 0$ and $\mathbf{f}^{\mathrm{N}} = \mathbf{0}$). The cylinder has a no-slip boundary condition. We use the far-field values to define a Reynolds number $\mathrm{Re} \coloneqq D\,(\rho u/\mu)_\infty$.

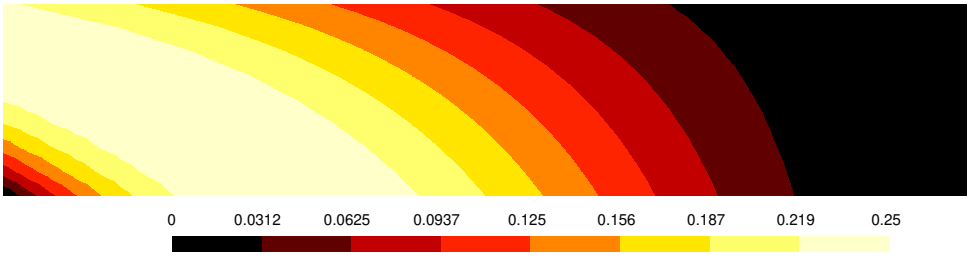
The initial condition requires special care. First, it is not easy to find an initial velocity field that satisfies $\mathbf{n}\cdot\mathbf{u} = \mathbf{n}\cdot\mathbf{u}^{\mathrm{D}}$ at the walls, inlet, and cylinder. We therefore initialize the velocity to $[u_\infty, 0]$, and leave out the convective term in the first 10 time steps, thus essentially simulating Stokes flow, which does not have the same



(a) u_1.



(b) u_2.



(c) Kinematic viscosity (ν) and thermal diffusivity (α). (Note that $\mathrm{Pr} = 1$, so $\alpha = \nu$.)

Figure 3.3: Constant-density manufactured solution in Eq. 3.16 at $t = 1$.

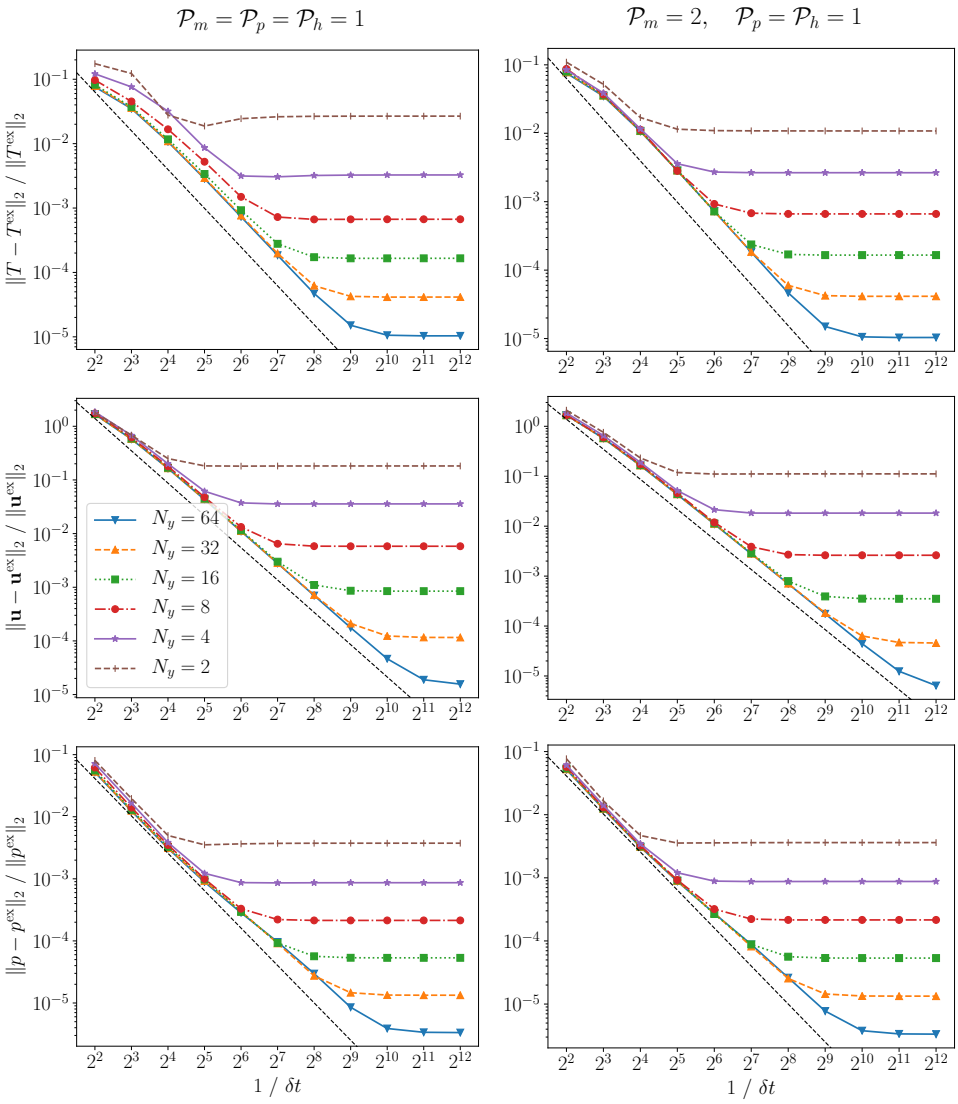


Figure 3.4: Convergence of the numerical solution toward the constant-density manufactured solution (Eq. 3.16) with temporal refinement. The characteristic element length is inversely proportional to N_y. The black dashed lines indicate ideal second-order convergence in δt.

Table 3.3: Convergence toward the constant-density manufactured solution in Eq. 3.16 (Fig. 3.3) with spatial refinement and fixed $\delta t = 2^{-12}$.

	temperature		velocity		pressure	
N_y	error	conv	error	conv	error	conv
Equal order $(\mathcal{P}_m = \mathcal{P}_p = \mathcal{P}_h = 1)$*:*						
2^1	2.67e-2		1.82e-1		3.75e-3	
2^2	3.26e-3	3.03	3.58e-2	2.35	8.66e-4	2.11
2^3	6.70e-4	2.29	5.81e-3	2.62	2.14e-4	2.02
2^4	1.66e-4	2.01	8.46e-4	2.78	5.34e-5	2.00
2^5	4.15e-5	2.00	1.16e-4	2.87	1.33e-5	2.00
2^6	1.04e-5	2.00	1.57e-5	2.89	3.34e-6	2.00
Mixed order $(\mathcal{P}_m = 2,\ \mathcal{P}_p = \mathcal{P}_h = 1)$*:*						
2^1	1.08e-2		1.11e-1		3.62e-3	
2^2	2.65e-3	2.03	1.83e-2	2.61	8.75e-4	2.05
2^3	6.63e-4	2.00	2.62e-3	2.80	2.15e-4	2.02
2^4	1.66e-4	2.00	3.52e-4	2.90	5.35e-5	2.01
2^5	4.15e-5	2.00	4.57e-5	2.94	1.34e-5	2.00
2^6	1.04e-5	2.00	6.43e-6	2.83	3.34e-6	2.00

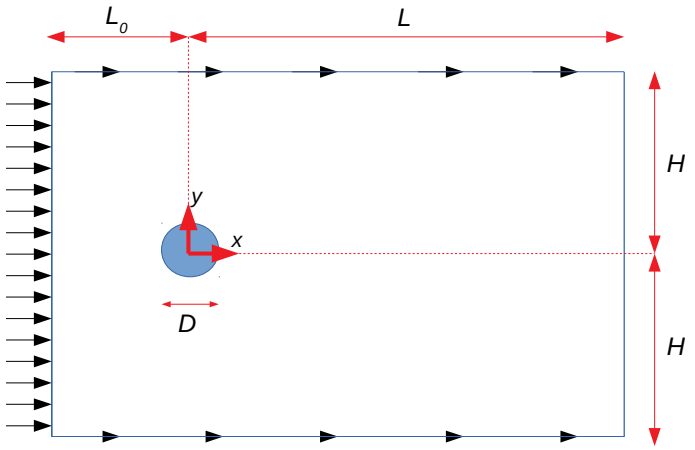


Figure 3.5: Geometry of flow past a circular cylinder (not to scale). The cylindrical obstacle is centered at the origin.

requirement for the initial condition at the boundaries. Due to the instant smoothing property of the viscous operator, the velocity satisfies the Dirichlet boundary conditions when the convective term is 'activated' after the first 10 time steps. This causes an instantaneous change in the pressure, but that is neither unphysical in the incompressible flow limit, nor is it a problem for the pressure correction method. Second, we found that the combination of a symmetrical mesh and a symmetrical initial condition does not induce vortex shedding. This is addressed by letting the cylinder rotate counter-clockwise for the first 100 time steps.

We are interested in the force on the cylinder S, which is given by

$$\mathbf{F}^{(\text{cyl})} = -\int_{\partial S} (\tau - p\ I) \cdot \mathbf{n}\ , \tag{3.17}$$

where $\mathbf{n}$ is the outward normal of the fluid, pointing into the cylinder. The drag and the lift coefficients are

$$C_\text{D} = \frac{2\ F_1^{(\text{cyl})}}{D\ (\rho u^2)_\infty} \quad \text{and} \quad C_\text{L} = \frac{2\ F_2^{(\text{cyl})}}{D\ (\rho u^2)_\infty} \tag{3.18}$$

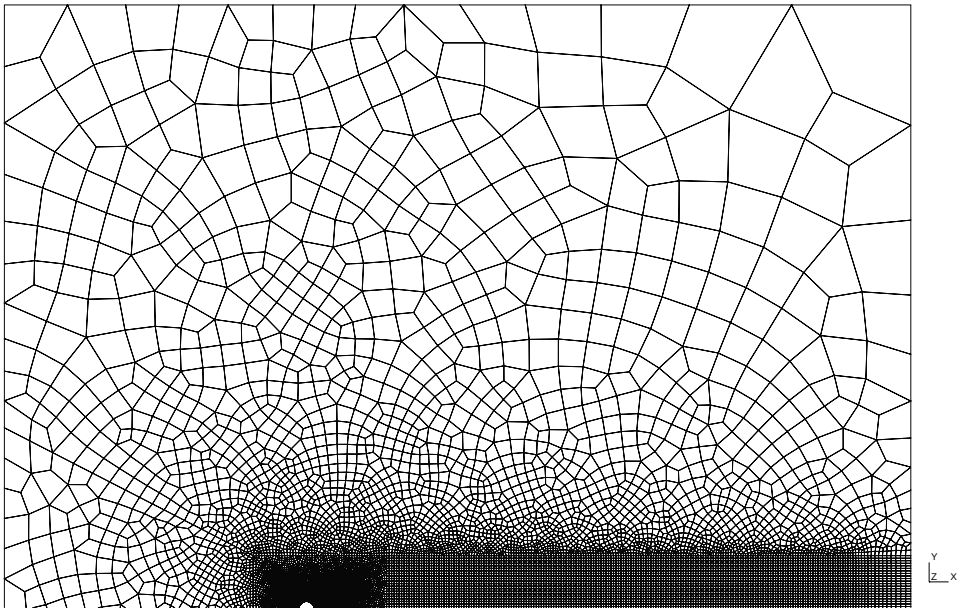
respectively. At our Reynolds number, the flow is laminar, and the force oscillates in time in a smooth, deterministic manner. This makes it easy to determine the frequency f of the lift coefficient, and the corresponding Strouhal number $\text{St} := fD/u_\infty$.

Our numerical experiments indicate that a small domain results in an overestimation of the Strouhal number. This likely explains the large discrepancy in the numerical predictions of the Strouhal number in previous literature; see Niroobakhsh *et al.* [21] for an overview. Collis [22] showed that a domain of $(H, L_0, L) = (30D, 15D, 30D)$ was sufficient for isothermal flow at $\text{Re} = 100$ and a Mach number of 0.2. The results presented here were obtained on a domain of $(H, L_0, L) = (40D, 20D, 40D)$. The domain could probably be substantially smaller if the upper and lower part were connected with periodic boundary conditions, but `DGFlows` did not support this yet when the calculations were done.

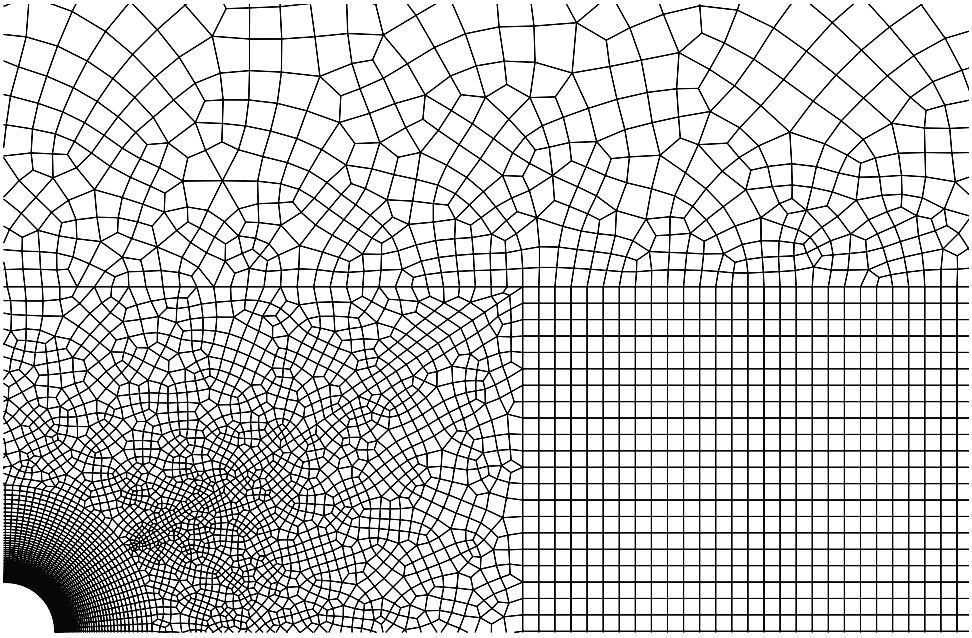
Fig. 3.6 shows the mesh. It has approximately 27k elements with $\mathcal{P}_m = 2$ and $\mathcal{P}_p = \mathcal{P}_h = 1$, resulting in approximately 160k degrees of freedom per direction of the mass flux, and 64k degrees of freedom for p and h. There are 120 boundary elements at the cylinder, each with a width of $0.005D$. The time step is given by $D/(\delta t\ u_\infty) = 82$. We find $\text{St} = 0.166$. This number remained unchanged when we decreased the number of elements to 19k, or when we doubled δt. It compares well with the experimental values of $\text{St} = 0.165$ in [23], $\text{St} = 0.165$ in [24], and $\text{St} = 0.167$ in [25, p. 71].

3.5. Discussion

The tests with the manufactured solutions in section 3.3 show second-order temporal accuracy in the L^2-norm for the velocity and the temperature. This observation is based on simulations on many levels of spatial discretization, and time step sizes



(a) overview (bottom not shown)



(b) Detail near the cylinder.

Figure 3.6: Mesh for flow past an obstacle. It is structured near the cylinder and most of its wake. The rest of the mesh is unstructured to allow for large differences in the element size. It is symmetrical about the axis $y = 0$.

that span many orders of magnitude (ref. Figs. 3.1 and 3.4). Even the pressure is not far from second-order accurate in δt.

An important and obvious improvement is the rotational formulation of the pressure correction method, as explained in section 3.2. Even though this requires two poisson solves per iteration in the case of a nonhomogeneous viscosity, previous works suggests that the improved accuracy would be worth the cost.

3.5.1. Pressure Correction with Equal-order Discretizations

The manufactured solutions show that the equal-order discretization is stable in the limit of small time steps. To achieve this, the pressure Poisson equation 3.9 has been modified with a penalty matrix $\boldsymbol{C}$.

Unfortunately this stabilization makes the linear system more expensive. This is in part because adding penalization raises the condition number, which can be substantial in turbulent flow, given the δt^{-1} scaling in Eq. 3.9, and the ν^{-1} scaling of the penalty parameter in Eq. 2.9. Perhaps more importantly, $\boldsymbol{C}$ depends on the viscosity, and so the pressure matrix is generally not constant, which has a large impact on the total computation time in our implementation `DGFlows`.

A large penalty parameter may be less detrimental to the efficiency of other solvers if they are suitably preconditioned. For example, Fehn *et al.* [26] have investigated several p-multigrid methods for a DG-SIP discretization of the poisson equation with a matrix-free implementation. They found that the most effective approach is to project the finest DG grid onto a continuous Galerkin space, which can in turn be coarsened with lower-order continuous solution spaces. The resulting solver was robust with respect to the SIP penalty parameter.

It should also be noted that the pressure stabilization $\boldsymbol{C}$ was developed for steady Stokes flow, and it is overly restrictive for time-dependent convecting flow. In our transient calculations, there are already two effects that implicitly stabilize the pressure. These could be used to reduce the penalty parameter in Eq. 2.9, or possibly to leave it out entirely.

The first stabilizing effect is inherent to the pressure correction method. The error due to time-splitting methods can be shown to be equivalent to perturbing the continuity equation with a pressure diffusion term $(\nabla^2 p)$ that is proportional to δt. This means that the inf-sup instability does not manifest itself at large time steps, as has been well documented for several spatial discretizations (e.g., [8, 27, 28]). Ferrer *et al.* [20, 29] have provided estimates for the minimal time step size in a DG discretization, and showed that this can still be higher than the maximum time step that satisfies the CFL condition.

The other source of pressure stability is due to the SIP discretization for the pressure matrix (Eq. 3.11), which contains a penalty term with a penalty parameter as in Eq. 2.16 with a diffusion parameter $K = 1$. Shahbazi [30, pp. 48–65] has successfully used the SIP pressure matrix with an equal-order discretization without extra pressure stabilization (i.e., $\boldsymbol{C} = 0$). Our tests (not shown here) also indicate that, for equal-order discretizations without pressure stabilization, the LDG pressure matrix is unstable for all reasonable time steps, whereas using the SIP matrix is feasible for a wide range of practical time step sizes, though it always becomes

unstable in the limit $\delta t \to 0$.

This provides an extra incentive for replacing the pressure LDG matrix by an SIP matrix (Eq. 3.11). It was originally motivated by its smaller stencil and lower condition number [5], but is could well be that its greater stability is more important.

References

[1] A. Hennink, M. Tiberga, and D. Lathouwers, *A Pressure-based solver for low-Mach number flow using a discontinuous Galerkin method,* Journal of Computational Physics **425**, 109877 (2021).

[2] C. M. Klaij, J. J. W. van der Vegt, and H. van der Ven, *Space-time discontinuous Galerkin method for the compressible Navier-Stokes equations,* Journal of Computational Physics **217**, 589 (2006).

[3] S. Rhebergen, B. Cockburn, and J. J. W. van der Vegt, *A space-time discontinuous Galerkin method for the incompressible Navier-Stokes equations,* Journal of Computational Physics **233**, 339 (2013).

[4] V. Ambati and O. Bokhove, *Space-time discontinuous galerkin discretization of rotating shallow water equations,* Journal of Computational Physics **225**, 1233 (2007).

[5] K. Shahbazi, P. F. Fischer, and C. R. Ethier, *A high-order discontinuous Galerkin method for the unsteady incompressible Navier-Stokes equations,* Journal of Computational Physics **222**, 391 (2007).

[6] B. Klein, B. Müller, F. Kummer, and M. Oberlack, *A high-order discontinuous Galerkin solver for low Mach number flows,* International Journal for Numerical Methods in Fluids **81**, 489 (2016).

[7] B. Krank, N. Fehn, W. A. Wall, and M. Kronbichler, *A high-order semi-explicit discontinuous galerkin solver for 3d incompressible flow with application to dns and les of turbulent channel flow,* Journal of Computational Physics **348**, 634 (2017).

[8] J. L. Guermond, P. Minev, and J. Shen, *An overview of projection methods for incompressible flows,* Computer Methods in Applied Mechanics and Engineering **195**, 6011 (2006).

[9] F. Saleri and A. Veneziani, *Pressure Correction Algebraic Splitting Methods for the Incompressible Navier-Stokes Equations,* SIAM Journal on Numerical Analysis **43**, 174 (2005).

[10] A. J. Chorin, *A numerical method for solving incompressible viscous flow problems,* Journal of Computational Physics **2**, 12 (1967).

[11] M. O. Henriksen and J. Holmen, *Algebraic Splitting for Incompressible Navier-Stokes Equations,* Journal of Computational Physics **175**, 438 (2002).

[12] B. Klein, *A high-order Discontinuous Galerkin solver for incompressible and low-Mach number flows*, Ph.D. thesis, Technische Universität, Darmstadt (2015).

[13] P. M. Gresho and R. L. Sani, *On pressure boundary conditions for the incompressible Navier-Stokes equations,* International Journal for Numerical Methods in Fluids **7**, 1111 (1987).

[14] J. L. Guermond and J. Shen, *On the error estimates for the rotational pressure-correction projection methods,* Mathematics of Computation **73**, 1719 (2004).

[15] L. J. P. Timmermans, P. D. Minev, and F. N. Van de Vosse, *An approximate projection scheme for incompressible flow using spectral elements,* International Journal for Numerical Methods in Fluids **22**, 673 (1996).

[16] J. Aoussou, J. Lin, and P. F. J. Lermusiaux, *Iterated pressure-correction projection methods for the unsteady incompressible Navier-Stokes equations,* Journal of Computational Physics **373**, 940 (2018).

[17] M. Piatkowski, S. Müthing, and P. Bastian, *A stable and high-order accurate discontinuous Galerkin based splitting method for the incompressible Navier-Stokes equations,* Journal of Computational Physics **356**, 220 (2018).

[18] J. Deteix and D. Yakoubi, *Improving the pressure accuracy in a projection scheme for incompressible fluids with variable viscosity,* Applied Mathematics Letters **79**, 111 (2018).

[19] Wikipedia contributors, *Kahan summation algorithm,* (2020), Wikipedia, The Free Encyclopedia. [Online] Accessed on 2020-09-14.

[20] E. Ferrer, D. Moxey, R. H. J. Willden, and S. J. Sherwin, *Stability of Projection Methods for Incompressible Flows Using High Order Pressure-Velocity Pairs of Same Degree: Continuous and Discontinuous Galerkin Formulations,* Communications in Computational Physics **16**, 817 (2014).

[21] Z. Niroobakhsh, N. Emamy, R. Mousavi, F. Kummer, and M. Oberlack, *Numerical investigation of laminar vortex shedding applying a discontinuous Galerkin finite element method,* Progress in Computational Fluid Dynamics, An International Journal **17**, 131 (2017).

[22] S. S. Collis, *Discontinuous Galerkin methods for turbulence simulation,* in *Summer Program 2002, Center for Turbulence Research* (2002) pp. 155–167.

[23] A.-B. Wang, Z. Trávníček, and K.-C. Chia, *On the relationship of effective Reynolds number and Strouhal number for the laminar vortex shedding of a heated circular cylinder,* Physics of Fluids **12**, 1401 (2000).

[24] C. H. K. Williamson, *Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low Reynolds numbers,* Journal of Fluid Mechanics **206**, 579 (1989).

[25] A. Roshko, *On the development of turbulent wakes from vortex streets*, Phd thesis, California Institute of Technology (1952).

[26] N. Fehn, P. Munch, W. A. Wall, and M. Kronbichler, *Hybrid multigrid methods for high-order discontinuous Galerkin discretizations,* Journal of Computational Physics **415**, 109538 (2020).

[27] J.-L. Guermond and L. Quartapelle, *On stability and convergence of projection methods based on pressure Poisson equation,* International Journal for Numerical Methods in Fluids **26**, 1039 (1998).

[28] N. Fehn, W. A. Wall, and M. Kronbichler, *On the stability of projection methods for the incompressible Navier-Stokes equations based on high-order discontinuous Galerkin discretizations,* Journal of Computational Physics **351**, 392 (2017).

[29] E. Ferrer and R. H. J. Willden, *A high order Discontinuous Galerkin Finite Element solver for the incompressible Navier-Stokes equations,* Computers & Fluids **46**, 224 (2011).

[30] K. Shahbazi, *A Parallel High-Order Discontinuous Galerkin Solver For the Unsteady Incompressible Navier-Stokes Equations in Complex Geometries*, Phd thesis, University of Toronto (2007).

4

Handling the Enthalpy Equation for Low-Mach Number Flow

4.1. Introduction

4.1.1. In Between Compressible and Incompressible

Several low-speed flows of practical importance are compressible, that is, the velocity is not divergence-free. This can occur due to mixing, or due to a temperature-dependent density near a heat source. An example is heat transfer in low-Mach number flows of supercritical fluids, where all fluid properties vary strongly with the temperature, but do not depend significantly on the pressure. Most flow solvers use either a pressure-based approach and assume a divergence-free velocity field, or a fully compressible (density-based) formulation. Neither of these methods is directly applicable to compressible flows in the low-Mach number limit.

Density-based solvers can be used to simulate zero-Mach flows by approximating the flow with a low, non-zero Mach number (e.g., [2], [3]). This has often been used for heat transfer in supercritical fluids at low speeds (e.g., [4], [5]). This is expensive for several reasons. First, the temporal discretization needs to resolve acoustic effects, and the resulting linear systems tend to be very stiff. Second, the system of transport equations is solved in a coupled way, which is more expensive than using a time-splitting method, though the performance may be improved with suitable preconditioning [6]. Finally, the fluid properties are evaluated as a function of two thermodynamic variables (usually the density and the volumetric enthalpy), so that a spline interpolation costs far more memory, thus complicating massively parallel calculations [4].

Parts of this chapter have been published in [1]

There is also substantial experience with discontinuous Galerkin (DG) discretizations for incompressible flows. These are either based on the introduction of artificial compressibility (e.g. [7, 8]), or they solve for the pressure (e.g. [9–12]). The artificial compressibility method can be more than second-order accurate in time, though it requires the system of transport equations to be solved in a coupled manner (e.g., [7, 13]). By choosing entropy variables as the unknowns, the DG method can also be formulated in a general way for both compressible and incompressible flows, at the cost of great complexity (e.g., [14]). There is, however, almost no literature on solving the low-Mach number equations with a pressure-based discontinuous Galerkin method, as is done in this thesis.

The only previous work of which we are aware is by Klein *et al.* [15, 16], who used a SIMPLE scheme to march the transport equations forward in time, iterating the equations within each time step. This required under-relaxation in order for the iteration to converge. They solved for the velocity, so that a predictor for the density is needed in the temporal derivative of the momentum equation.

We avoid this by solving for the mass flux rather than the velocity. Another advantage of this approach is that the divergence term in the continuity equation does not have to be weighed by the density, so that the divergence matrix does not depend on the density. This makes the transport equations less tightly coupled, and it simplifies the pressure correction method, because the pressure matrix is constant for each time step.

4.1.2. Which Enthalpy Equation Should be Solved?

Another important question is which form of the enthalpy transport equation should be solved (primitive or conservative), and for which variable (the primitive h or conserved $H = \rho h$).

Solving for the specific enthalpy h from the primitive transport equation, that is,

$$\frac{\mathrm{D}h}{\mathrm{D}t} \coloneqq \frac{\partial h}{\partial t} + \mathbf{u} \cdot \nabla h = \frac{1}{\rho} \nabla \cdot \left(\frac{k}{c_p} \nabla h \right) + \frac{1}{\rho} Q \ , \tag{4.1}$$

would pose two possible problems. First, the diffusive term is not in the standard form $\nabla \cdot (\alpha \nabla h)$ where $\alpha \coloneqq k / \left(\rho c_p \right)$ is the thermal diffusivity. The difference is

$$\frac{1}{\rho} \nabla \cdot \left(\frac{k}{c_p} \nabla h \right) - \nabla \cdot (\alpha \nabla h) = - \left(\nabla \frac{1}{\rho} \right) \cdot \frac{k}{c_p} \nabla h = \alpha \ \mathbf{d} \cdot \nabla h \ , \tag{4.2}$$

where

$$\mathbf{d} \coloneqq \frac{1}{\rho} \nabla \rho = \frac{\rho_h}{\rho} \nabla h \tag{4.3}$$

is the relative gradient of the density (as in Eq. 2.13). Perhaps the commutation error in Eq. 4.2 can be neglected in many flows. It is also possible that the diffusion term in Eq. 4.1 can be discretized by a standard interior penalty method with a diffusion parameter of k/c_p, except that the test function is weighed by $1/\rho$.

A more fundamental problem with the primitive transport equation 4.1 is posed by the advection term. It is not in conservative form, as in Eq. 2.18, and so we do

not know how to discretize it with a DG method. Perhaps it could be split as

$$\mathbf{u} \cdot \nabla h = \nabla \cdot (\mathbf{u}\ h) - h \nabla \cdot \mathbf{u}\ . \tag{4.4}$$

The first term on the right is in standard form, whereas the second term resembles the divergence of the mass flux in the continuity equation, but we are not sure how to proceed from there. We therefore solve the transport equation in the conservative form 1.1c.

For high-Mach number compressible flows, the unknown is normally taken to be a conserved variable, such as the volumetric enthalpy $H \coloneqq \rho h$. When written in that variable, the transport equation becomes

$$\frac{\partial H}{\partial t} + \nabla \cdot \left(\frac{1}{\rho} \mathbf{m}\ H \right) = \nabla \cdot \left(\frac{k}{c_p} \nabla \left(\frac{H}{\rho} \right) \right) + Q\ . \tag{4.5}$$

The diffusion term is again not in the standard form (due to the factor of $1/\rho$), in direct analogy to when the viscous stress tensor is expressed in terms of a conserved variable (m), as was discussed in section 2.3.1. The generalization of the symmetric interior penalty method for the viscous stress could also be applied here. That is, one starts from a standard discretization for $\nabla \cdot (\alpha \nabla H)$, and wherever the stress $\alpha \nabla H$ would appear in the weak form, it is replaced by $\alpha\ (\nabla H + H\mathbf{d})$.

Incidentally, there is a curious alternative interpretation of the effect of the variable density on the thermal diffusion, namely that it results in an extra advective flux. Substituting Eq. 4.2 into Eq. 4.1 gives

$$\frac{\partial h}{\partial t} + \mathbf{u}^{+}_{\text{eff}} \cdot \nabla h = \nabla \cdot (\alpha \nabla h) + \frac{1}{\rho} Q\ , \tag{4.6}$$

where

$$\mathbf{u}^{\pm}_{\text{eff}} = \mathbf{u} \mp \alpha \mathbf{d} = \mathbf{u} \pm \beta \alpha \nabla T \tag{4.7}$$

is the effective advecting velocity, and $\beta \coloneqq -(1/\rho)\rho_T$ is the thermal expansibility. Similarly, the conservative transport equation 4.5 has a diffusion term with commutation error

$$\nabla \cdot \left(\frac{k}{c_p} \nabla \left(\frac{H}{\rho} \right) \right) - \nabla \cdot (\alpha \nabla H) = -\nabla \cdot (H \alpha \mathbf{d}) = \nabla \cdot (H \alpha \beta \nabla T)\ , \tag{4.8}$$

so that

$$\frac{\partial H}{\partial t} + \nabla \cdot \left(\mathbf{u}^{-}_{\text{eff}} H \right) = \nabla \cdot (\alpha \nabla H) + Q\ . \tag{4.9}$$

(Note the different signs of the advection correction $\pm \alpha \beta \nabla T$ in Eqs. 4.6 and 4.9.)

This observation is probably only of theoretical significance, though the equations with an adjusted advecting velocity could also be used for the discretization. The extra term $\beta \alpha \nabla T$ can treated explicitly (i.e., extrapolated from previous time steps). Since $(\mathbf{u}_{\text{eff}} H - \alpha \nabla H) \cdot \mathbf{n} = \left(\mathbf{u}\ H - \left(k/c_p \right) \nabla h \right) \cdot \mathbf{n}$, the boundary contributions in the weak forms are the same as for the regular equation 1.1c. Our brief tests

(not shown here) suggest that this is feasible, and second-order temporal accuracy for the enthalpy is maintained in the L^2-norm.

Coming back to the question of which equation to solve for which unknown, the main problem with the volumetric enthalpy is that it is not unique to a particular thermodynamic state, which will be a recurring theme in this chapter. The combination of the specific enthalpy h (or, equivalently, the temperature) and the thermodynamic pressure p^{th} fixes the thermodynamic state, but the pair $\left(H, p^{\mathrm{th}}\right)$ does not always do this, making H an inconvenient thermodynamic variable. This problem is not encountered by compressible flow solvers at high Mach numbers, because they solve a transport equation for both H and ρ, which implies a specific enthalpy h from which the fluid properties can be derived.

For example, consider an ideal gas with a specific gas constant r:

$$p^{\mathrm{th}} = r\rho T \ , \tag{4.10}$$

for which $\rho_T = -\rho/T$, and thus $H_h = \rho + h\rho_T/c_p = \rho\left(1 - h/\left(c_p T\right)\right)$. If $h = c_p T$, as is sometimes assumed, then $H_T = c_p H_h = 0$, meaning that the value of H says nothing about the temperature.

Another example can be found in fluids at a supercritical temperature. If the temperature is increased from a point close to a liquid state, then the volumetric enthalpy will rise initially, but then it will drop when the so-called Widom line is crossed, where the thermal expansibility peaks [17]. In other words H_T switches sign, and thus H does not uniquely determine T.

Solving for the volumetric enthalpy would therefore require some special treatment to determine the thermodynamic state in low-Mach number flows. One possibility is to offset the specific enthalpy by a suitable constant, which would produce a one-to-one relation between the temperature and the new volumetric enthalpy, as will be explained in section 4.3 (see Fig. 4.1 in particular). Another idea would be to obtain a separate predictor for the density, for example by solving the continuity equation, so that the specific enthalpy can be estimated.

Peeters [18] has claimed to have taken this approach, though we note that the nondimensionalization of the variables in that work also involved offsetting the specific enthalpy by a constant. It is possible that this created a one-to-one relationship between the nondimensional h and the nondimensional H, in which case they would have inadvertently solved the non-uniqueness of H even before the density predictor was introduced.

We avoid these difficulties by solving for the specific enthalpy h from the conservative transport equation 1.1c, which poses another problem that needs to be overcome. Solving a conservative transport equation for a primitive variable complicates the temporal derivative: the term $\partial(\rho h)/\partial t$ can be an important source of error and instability when ρ is a function of h. A similar issue occurs in multispecies transport, and Najm *et al.* [19] devised a widely used two-step iterative method to stabilize the temporal scheme. This has subsequently been adapted to handle the strong property variations in supercritical fluids [20]. One could also obtain a density predictor by solving the continuity equation.

In sections 4.3 and 4.4 we present a new alternative method that does not use

any predictor solves or iterations to handle the unknown density at a new time step. We will show that the error in our approximation can be made negligible compared to the error in the finite difference scheme, and that the method can be made unconditionally stable by offsetting the specific enthalpy with a constant.

4.2. The Temporal Density Gradient

The pressure-correction method for incompressible flow in section 3.2 can be modified for low-Mach number flow in a straightforward manner. The semi-discrete continuity equation 3.2a is adjusted to include the temporal derivative of the density:

$$-\boldsymbol{D}\underline{\boldsymbol{m}}^n + \boldsymbol{C}\underline{\boldsymbol{p}}^n = -\underline{\boldsymbol{r}} - \mathcal{G}\left[\left(\frac{\partial \rho}{\partial t}\right)^n\right] =: -\underline{\tilde{\boldsymbol{r}}}^n \,, \tag{4.11}$$

where $\mathcal{G}[\cdot]$ denotes the Galerkin projection onto the solution space. The temporal derivative of the density is estimated with a second-order backward finite difference scheme:

$$\left(\frac{\partial \rho}{\partial t}\right)^n \approx \frac{1}{\delta t}\left(\frac{3}{2}\rho^n - 2\rho^{n-1} + \frac{1}{2}\rho^{n-2}\right) \,. \tag{4.12}$$

The time-splitting scheme is exactly the same as in section 3.2, except that $\underline{\boldsymbol{r}}$ is replaced by $\underline{\tilde{\boldsymbol{r}}}^n$.

This is a large advantage of solving for the mass flux. If we had instead solved for the velocity (such as in [15]), then the density would have had to have been incorporated into the divergence operator D in Eq. 3.9, and into the mass matrix M in Eq. 3.4.

This extension of the pressure-correction method to compressible flows has sometimes proved unstable in finite difference schemes that were applied to mixing flows with large density ratios (of approximately more than a factor of 3), because the continuity equation was not satisfied in the inviscid limit; see Nicoud [21]. It is not certain whether the same instability would occur for the discontinuous Galerkin method presented here; our experience so far has not exposed instabilities with large density ratios. Nicoud suggested a different generalization of the pressure correction method to variable-density flows, where the density is incorporated into the pressure matrix, rather than on the right-hand side of Eq. 3.9.

The large advantage of the approach presented here is that the pressure matrix is the same at all time steps.[1] We can therefore assemble it once, and precompute the incomplete Cholesky preconditioner for the linear solver. Furthermore, the condition number of the diffusion matrix A worsens if it includes a variable coefficient that depends on the density. For these reasons, the pressure solves are much cheaper with a constant pressure matrix.

Irrenfried [22, 23] also claimed (without reference or demonstration) that using Eq. 4.12 may result in a considerable numerical error for small time steps, and

[1] except for equal-order discretizations with a temperature-dependent kinematic viscosity, in which case the pressure stabilization (Eqs. 2.8–2.9) depends on h

therefore used an interesting alternative, based on

$$\frac{\partial \rho}{\partial t} = \rho_T \frac{\partial T}{\partial t} \ . \tag{4.13}$$

The idea is to replace $\partial T/\partial t$ on the rhs by the spatial discretization of the temperature. This was probably straightforward to implement in their work, which is based on a finite volume scheme for the temperature equation in primitive form (i.e., $DT/Dt = \ldots$). The idea is certainly attractive: given the one-to-one correspondence between the density and the temperature (or the enthalpy), it seems natural to use the same spatial discretization for their temporal derivatives.

It is less clear how the equivalent approach could be used here for at least two reasons. First, $(\rho h)_\rho$ is not constant, and we use a finite element method, so we cannot simply multiply the solution vector by the value of a fluid property in a point-wise manner. This would probably have to be done with a Galerkin projection instead. A more fundamental problem is that we solve for the enthalpy equation in conservative form, so we have a spatial discretization for $\partial(\rho h)/\partial t$, not $\partial h/\partial t$. Unfortunately (ρh) does not uniquely determine ρ in many fluids, and therefore the relation

$$\frac{\partial \rho}{\partial t} = \frac{1}{(\rho h)_\rho} \frac{\partial (\rho h)}{\partial t} \tag{4.14}$$

could be undefined, or very badly conditioned, since $\left|(\rho h)_\rho\right|$ can be arbitrarily small. This problem of a non-unique volumetric enthalpy (ρh) will return later in this chapter. Perhaps Eq. 4.14 could be used if $(\rho h)_\rho$ is made non-zero by a suitable enthalpy offset, as in section 4.3.2.

4.3. Linearizing $\partial(\rho h)/\partial t$

As mentioned in the introduction, solving for a primitive variable (h) with the enthalpy equation in conservative form complicates the temporal derivative of the enthalpy, because it is weighed by the temperature-dependent density. This section and the next study the stability and convergence of the time stepping scheme in detail.

The analysis is simplified by considering a space-independent enthalpy equation:

$$\frac{\mathrm{d}\,(\rho h)}{\mathrm{d}t} = -\lambda h + Q \ , \tag{4.15}$$

where λ is a constant, and $Q = Q(t)$. Using an implicit finite difference scheme, the enthalpy and the corresponding density can be estimated at a time step n by

$$\frac{\gamma_0}{\delta t} (\rho h)^n + \sum_{i=1}^{q} \frac{\gamma_i}{\delta t} (\rho h)^{n-i} = -\lambda h^n + Q^n \ . \tag{4.16}$$

Due to the variable density, this equation is not linear in the unknown h^n. We therefore consider two linearizations in h^n, which we term method #1 and method #2.

Both of these methods use a predictor h^* and a corresponding ρ^* that are close to h^n and ρ^n. This predictor can be obtained in several ways, such as by extrapolating from previous time steps. When solving the full system 1.1a-1.1c, a predictor for ρ^n can also be obtained by solving the continuity equation. The analyses in this section are for a general (h^*, ρ^*), though we will make the reasonable assumption that $(h^* - h^n)$ is at least first-order accurate in δt.

The two linearization methods are as follows.

Method #1 is perhaps the most obvious approach: let $\rho^n \approx \rho^*$, resulting in an approximation $h^{[1]} \approx h^n$ that is given by

$$\frac{\gamma_0}{\delta t}\rho^* h^{[1]} + \sum_{i=1}^{q} \frac{\gamma_i}{\delta t} (\rho h)^{n-i} = -\lambda h^{[1]} + Q^n \ . \tag{4.17}$$

Method #2 is based on a Taylor expansion of $(\rho h)^n$ about the predictor:

$$(\rho h)^n \approx (\rho h)^* + (\rho h)^*_h (h^n - h^*) = (\rho h)^*_h h^n - \left(h^2 \rho_h\right)^* \ . \tag{4.18}$$

Substituting this into Eq. 4.16 yields an approximation $h^{[2]} \approx h^n$, given by

$$\frac{\gamma_0}{\delta t}(\rho h)^*_h h^{[2]} + \sum_{i=1}^{q} \frac{\gamma_i}{\delta t} (\rho h)^{n-i} = \frac{\gamma_0}{\delta t}\left(h^2 \rho_h\right)^* - \lambda h^{[2]} + Q^n \ . \tag{4.19}$$

Note that method #1 is effectively a single step in a fixed-point iteration, whereas method #2 is a single step in a Newton iteration.

4.3.1. Error Estimates and Stability

The errors and the stability of methods #1 and #2 can be analyzed by using a Taylor series for ρ^n about the predictor, that is,

$$\rho^n = \sum_{k=0}^{\infty} \frac{1}{k!} \left(\frac{\partial^k \rho}{\partial h^k}\right)^* (h^n - h^*)^k \ . \tag{4.20}$$

Define the following deviations from the non-linear finite difference equation 4.16:

$$\begin{aligned} \text{error in the predictor:} \quad & \epsilon^* \coloneqq h^* - h^n \ , \\ \text{linearization error in method \#1:} \quad & \epsilon^{[1]} \coloneqq h^{[1]} - h^n \ , \\ \text{linearization error in method \#2:} \quad & \epsilon^{[2]} \coloneqq h^{[2]} - h^n \ . \end{aligned} \tag{4.21}$$

The derivations are tedious, and deferred the appendix at the end of this chapter. Here we summarize the main theoretical results.

The first result is an a priori error estimate. Appendix A shows that both Eq. 4.17 and Eq. 4.19 can be rewritten as

$$\frac{\gamma_0}{\delta t} (\rho h)^n + \sum_{i=1}^{q} \frac{\gamma_i}{\delta t} (\rho h)^{n-i} = -\lambda_{\text{eff}}\, h^n + Q^n_{\text{eff}} \ , \tag{4.22}$$

where

$$\lambda_{\text{eff}} = \lambda + \mathcal{O}\left(\epsilon^{*2}/\delta t\right) \quad \text{and} \quad Q^n_{\text{eff}} = Q^n + \mathcal{O}\left(\epsilon^{*2}/\delta t\right) \quad \text{for method \#1,} \tag{4.23a}$$

and

$$\lambda_{\text{eff}} = \lambda + \mathcal{O}\left(\epsilon^{*3}/\delta t\right) \quad \text{and} \quad Q^n_{\text{eff}} = Q^n + \mathcal{O}\left(\epsilon^{*3}/\delta t\right) \quad \text{for method \#2.} \tag{4.23b}$$

That is, the approximations in method #1 and #2 are equivalent to the original Eq. 4.16, except that λ and Q^n are replaced by their effective values, which are related to the error in the predictor.

A second important result regards the stability of the linearization methods. Appendix A.1 shows that the error for method #1 is related to the error in the predictor as

$$\begin{aligned} \frac{\epsilon^{[1]}}{\epsilon^*} &= -\left(\frac{\rho_h h}{\rho + \lambda\,(\delta t/\gamma_0)}\right)^* + \mathcal{O}\,(\epsilon^*) \\ &= -\left(\frac{\rho_h h}{\rho}\right)^* + \mathcal{O}\,(\delta t) + \mathcal{O}\,(\epsilon^*) \end{aligned} \tag{4.24}$$

Note that $\epsilon^{[1]}/\epsilon^*$ vanishes up to first order as $h^* \to 0$. Eq. 4.24 also suggests that method #1 cannot always be made stable by iterating within a time step, that is, by calculating a new predictor ρ^* from the estimate $h^{[1]}$, and repeating Eq. 4.17. Stability of the iteration is guaranteed if the error in the new approximation is always smaller than the error in the predictor, that is, $\left|\epsilon^{[1]}\right| < |\epsilon^*|$. This condition is only met if

$$|h^*| < \left|-\frac{\rho}{\rho_h}\right|^* = \left(\frac{c_p}{\beta}\right)^* , \tag{4.25}$$

where we have made the reasonable assumptions that $\epsilon^* = \mathcal{O}\,(\delta t)$, and that $\rho_h < 0$. If Eq. 4.25 is not met at every step in the iteration, then it may not converge.

Similarly, iterating method #2 within a time step is stable if $\left|\epsilon^{[2]}\right| < |\epsilon^*|$. Appendix A.2 shows that

$$\left((\rho h)^*_h + \frac{\delta t}{\gamma_0}\lambda\right)\frac{\epsilon^{[2]}}{\epsilon^*} = \left(\rho_h + \frac{1}{2}\rho_{hh} h\right)^* \epsilon^* + \mathcal{O}\left(\epsilon^{*2}\right) . \tag{4.26}$$

Since we can reasonably expect that the error in the predictor (ϵ^*) is at least first-order accurate in δt, we always have $\left|\epsilon^{[2]}\right| < |\epsilon^*|$ (and therefore a stable iteration) in the limit $\delta t \to 0$, provided that

$$(\rho h)^*_h \neq 0 . \tag{4.27}$$

In other words, the volumetric enthalpy (ρh) must be a strictly monotonic function of the specific enthalpy h.

This restriction for method #2 also follows more directly from Eq. 4.19 in the limit of small time steps, because the coefficient of $h^{[2]}$ cannot vanish. In practice one will want to satisfy the stronger relation

$$(\rho h)_h > 0 \tag{4.28}$$

to ensure that the enthalpy discretization is positive definite.

We conjecture that the stability requirements (Eq. 4.25 for method #1; Eq. 4.27 for method #2) must always be satisfied in the limit of small time steps, even when the linearization is not iterated within a time step. It seems reasonable to expect that a stable numerical method can be iterated without diverging. This is supported by the numerical tests in Section 4.4

4.3.2. Proper Scaling of the Enthalpy Equation

Curiously, the analyses in the previous subsection have led to stability requirements (Eqs. 4.25, 4.27) that depend on the fluid properties, and they are not satisfied for all fluids. For example, the volumetric enthalpy in supercritical fluids can either increase or decrease with the temperature due to the strong thermal expansion, thereby violating Eq. 4.27.

This problem can be addressed by solving for a different variable

$$\tilde{h} \coloneqq h - h_0 \ . \tag{4.29}$$

Eq. 1.1c then becomes

$$h_0 R + \frac{\partial(\rho\tilde{h})}{\partial t} + \nabla\cdot\left(\mathbf{m}\,\tilde{h}\right) = \nabla\cdot\left(\frac{k}{c_p}\nabla\tilde{h}\right) + Q \ , \tag{4.30}$$

where $R \coloneqq \partial\rho/\partial t + \nabla\cdot\mathbf{m} = 0$ is the residual of the continuity equation 1.1a. Thus $\tilde{h}$ satisfies the same transport equation as h, and it can be discretized in the same way. This does not affect the diffusion (since $\nabla\tilde{h} = \nabla h$), but it does change the convection and the temporal derivative, which now has a different stability guarantee.

In particular, Eq. 4.25 for method #1 becomes

$$\left|\tilde{h}^*\right| = \left|h^* - h_0\right| < \left(\frac{c_p}{\beta}\right)^* \ . \tag{4.31}$$

We cannot know in advance whether this will be satisfied at all time steps. There is therefore no a priori value for h_0 that guarantees stability, though it seems that h_0 is best chosen such that $h \approx h_0$ at the average temperature.

Conversely, we *can* find an a priori lower bound for h_0 when using method #2. The stability guarantee (Eq. 4.28) becomes

$$\left(\rho\tilde{h}\right)_{\tilde{h}} = \left(\rho(h-h_0)\right)_h = (\rho h)_h - h_0\rho_h > 0 \quad\Leftrightarrow\quad h_0 > (\rho h)_\rho = h - \frac{c_p}{\beta} \ , \tag{4.32}$$

and so method #2 can be made unconditionally stable by choosing h_0 sufficiently large. In particular, if the temperature is known to lie in a range $[T^{\min}, T^{\max}]$, then a theoretical lower bound for stable values for h_0 is

$$h_0^{\min} = \max_{T^{\min}\le T\le T^{\max}} (\rho h)_\rho = \max_{T^{\min}\le T\le T^{\max}} \left(h - \frac{c_p}{\beta}\right) \ , \tag{4.33}$$

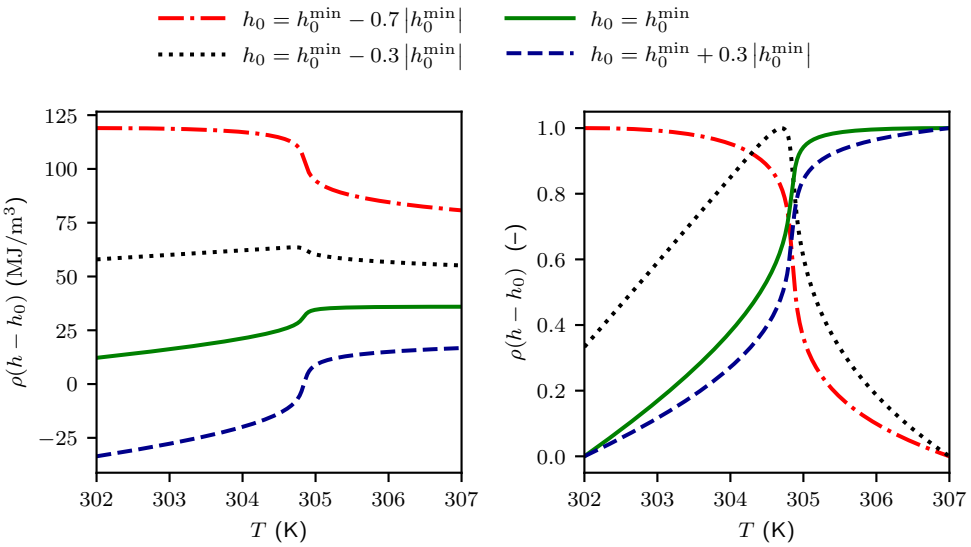


Figure 4.1: *Top*: rescaled volumetric enthalpy ($\rho(h-h_0)$) of carbon dioxide at the supercritical pressure of 7.5 MPa, as a function of the temperature for various choices of h_0. *Bottom*: same data, but with each line scaled to $[0,1]$.
The function increases monotonically for $h_0 \geq h_0^{\min}$. The thermodynamic reference point is placed at (1 bar, 0 °C). The data are based on [24], accessed through the `CoolProp` software library [25].

which can of course be negative. Fig. 4.1 shows an example of the rescaled volumetric enthalpy ($\rho(h - h_0)$) for various choices of h_0 for a real fluid in the temperature range $(302\mathrm{K}, 307\mathrm{K})$. There is a practical limit on the magnitude of h_0, because we are solving a transport equation for $\rho(h - h_0)$, which becomes equivalent to the density ρ for very large values of $|h_0|$.

On a heuristic level, the change of variables in Eq. 4.29 can be thought of as a way to discretize something in between the conservative and the primitive transport equations. The conservative transport equation is merely based on the conservation of enthalpy, whereas deriving the primitive transport also requires the conservation of mass. The primitive equation can be obtained by subtracting hR from the conservative equation. Comparing Eq. 4.30, we are subtracting $h_0 R$ from the conservative equation, and h_0 determines how much of the continuity equation is used for the enthalpy transport.

Of course a rescaling of the unknowns, such as in Eq. 4.29, is not unknown in CFD literature, but it has usually been presented as a mere numerical convenience (e.g., [18]). The above analyses show that the accuracy and stability of the numerical scheme depend critically on a proper choice of h_0. In practice this may require some trial and error, though these analyses offer useful guidelines.

4.3.3. Special Case of an Ideal Gas

Since many fluids are accurately described by the ideal gas law (Eq. 4.10), it is worth specializing the above analyses to this particular case. The specific heat

capacity is usually approximately constant in an ideal gas, so that $h = c_p T - h_0$ for some constant h_0. As already mentioned in section 4.1.2, solving the conservative enthalpy transport equation does not make sense when $h = c_p T$ (i.e., $h_0 = 0$), since the volumetric enthalpy would be constant. However, both method #1 and method #2 can be stable for a proper choice of a nonzero constant h_0.

First we consider the relationship between the linearization error ($\epsilon^{[1]}$) and the predictor error (ϵ^*) for method #1, given by Eq. 4.24. From the equation of state we have $\rho_h = -\rho/(c_p T)$. Eq. 4.24 becomes

$$\frac{\epsilon^{[1]}}{\epsilon^*} = \frac{h^*}{h^* + h_0} + \mathcal{O}(\delta t) + \mathcal{O}(\epsilon^*) \ . \tag{4.34}$$

Stability for method #1 is guaranteed if $\left|\epsilon^{[1]}/\epsilon^*\right| < 1$ for all time steps, which is equivalent to $h^*/h_0 > 1/2$.

For method #2, the volumetric enthalpy should be a strictly monotonically increasing function of the temperature. For an ideal gas, $H_T = \rho(c_p - h/T)$. In the case $h = c_p T - h_0$ this becomes $H_T = (\rho/T)h_0$, meaning that we must set $h_0 > 0$.

4.4. Test Case for the Space-independent Enthalpy Equation[2]

Before solving the full system of transport equations, we clarify the theoretical results for the space-independent enthalpy equation 4.15 in Section 4.3 with a numerical example that is based on a manufactured solution. Omitting the units of measurement, the exact temperature is

$$T^{\mathrm{ex}}(t) = 0.5 + 0.1\sin(2\pi t) \tag{4.35}$$

with $0 \le t \le 1$. The equation of state is

$$\rho = \rho_0 T + \rho_1(1 - T) \ , \tag{4.36}$$

and the specific heat capacity is kept constant, so that

$$h = c_p T - h_0 \ . \tag{4.37}$$

The required source term $Q(t)$ follows from Eq. 4.15. For the numerical tests we let $\rho_0 = 0.5$, $\rho_1 = 2$, $c_p = 1$, and $\lambda = 0.1$. The results presented here were all obtained at $\delta t = 2^{-11}$ to investigate the limit of small time steps. We have checked that lowering the time step size to $\delta t = 2^{-14}$ does not affect whether the numerical method is stable. To ensure that rounding errors did not play a significant role with these tiny time steps, all calculations in this section were performed with 128-bit floating point precision.

The numerical schemes were tested with various orders of the BDF time stepping scheme. The predictor h^* is obtained with an s^{th}-order extrapolation from previous

[2] The code for the finite difference method for the space-independent enthalpy equation can be found on GitHub [26]. It can be used to reproduce the results in this section.

Table 4.1: Coefficients for extrapolation from previous time steps. (See Eq. 4.38).

	α_1	α_2	α_3	α_4
EX1	1			
EX2	2	-1		
EX3	3	-3	1	
EX4	4	-6	4	-1

Table 4.2: Order of extrapolation for the enthalpy predictor (Eq. 4.38) for the linearization methods described in Section 4.3. The minimum values satisfy Eq. 4.39; from the maximum value onward, Eq. 4.39 holds with strict inequality.

finite difference coefficients	Method #1		Method #2	
	min	max	min	max
BDF1	EX2	EX2	EX1	EX2
BDF2	EX2	EX3	EX2	EX2
BDF3	EX3	EX3	EX2	EX2

time steps (denoted by EXs), and the corresponding ρ^* is determined from the equation of state. Specifically,

$$h^* = \sum_{i=1}^{s} \alpha_i h^{n-i} = h^n + \mathcal{O}\left(\delta t^s\right) \ . \tag{4.38}$$

The weights are in Table 4.1.

There are two numerical errors in each time step: (i) the BDF error, which is inherent in the finite difference scheme, and (ii) the linearization error in going from Eq. 4.16 to either Eq. 4.17 or Eq. 4.19 when using method #1 or method #2. If the EXs coefficients are used to obtain a predictor, then the error in the predictor is $\epsilon^* := h^* - h^n = \mathcal{O}\left(\delta t^s\right)$. For method #1, Eq. 4.23a then implies a linearization error of $\mathcal{O}\left({\epsilon^*}^2/\delta t\right) = \mathcal{O}\left(\delta t^{2s-1}\right)$. A BDF$q$ scheme makes an $\mathcal{O}\left(\delta t^{q+1}\right)$ error per time step, so that the overall order of accuracy is $\min(2s-1, q+1)$. Similarly, Eq. 4.23b implies that the overall error per time step for method #2 is of order $\min(3s-1, q+1)$. The order of extrapolation should therefore satisfy

$$s \geq \begin{cases} (q+2)/2 & \text{for method \#1,} \\ (q+2)/3 & \text{for method \#2,} \end{cases} \tag{4.39}$$

or else the linearization error dominates, and the usual order of convergence of the BDF scheme cannot be achieved. If strict inequality in Eq. 4.39 is satisfied, then the linearization error is negligible, and increasing the order of extrapolation is pointless. Table 4.2 lists the range of reasonable extrapolation orders.

Fig. 4.2 shows the error in the numerical temperature as a function of h_0, by using method #1 (Eq. 4.17). Note how the calculations diverge when h_0 is either

too small or too large. The stability guarantee in Eq. 4.25 cannot be determined a priori, because h^* is not known before the calculation. For low extrapolation order, method #1 sometimes converges even when Eq. 4.25 is not met at all time steps. According to Table 4.2, the linearization error is negligible compared to the BFD error for extrapolation orders of at least 2, 3, and 3 for the BDF1, BDF2, and BDF3 schemes. For these cases Eq. 4.25 becomes a strict requirement for stability. Note that the range of stable values for h_0 decreases with higher-order extrapolations, but all simulations converge with $h_0 = 0.5$, which is the value for which h is closest to zero.

Fig. 4.3 shows the equivalent error plots for method #2 (Eq. 4.19). For the current equation of state, we have an explicit, a priori expression for the stability criterion in Eq. 4.27:

$$(\rho h)_T \neq 0 \quad \Leftrightarrow \quad h_0/c_p \neq 2T - \rho_1/(\rho_1 - \rho_0) \; . \tag{4.40}$$

The tests show that the numerical scheme is stable if and only if this criterion is satisfied everywhere in the domain, regardless of the order of the time-stepping scheme, or the order of extrapolation for the predictor. Furthermore, the results show that the minimal extrapolation orders in Table 4.2 need to be reached in order to achieve the lowest errors, but higher orders of extrapolation have no effect.

4.5. Test Cases with Low-Mach Number Flow

4.5.1. Variable-density Manufactured Solution

As in the previous chapter, a manufactured solution is used to verify the numerical scheme and its implementation. The exact solution constructed by working backward from the exact mass flux and pressure, which can be chosen arbitrarily. The choice of the pressure is of little consequence, though we make sure that both m and p vary non-linearly in time, and that they do not lie in the numerical solution space. Integrating the continuity equation over time then gives the density, which in turn determines the temperature and the enthalpy. The transport properties (μ and k) are arbitrary functions of the temperature. The external force and heat source follow from Eqs. 1.1b and 1.1c.

It is surprising that we could not find previous work with a manufactured solution that is (i) compressible, (ii) uses temperature-dependent transport properties, and (iii) satisfies the unmodified continuity equation (without an artificial mass source). Most previous work has focussed on finding clever analytical solutions to the system 1.1 with a variable density (see, e.g., [27], and the references therein). Others (e.g., [28]) have included a source term in the continuity equation, but this appears less suitable for a time-splitting method, where the continuity plays a central role in the discretization (as in Section 4.3.2), and we do not want to adapt the numerical scheme to conform with the manufactured solution. Perhaps the current approach was not taken before because it results in non-trivial source terms (F and Q). We handle these calculations symbolically with the Python SymPy library. The manufactured solution is made up of polynomials to keep these symbolic calculations feasible.

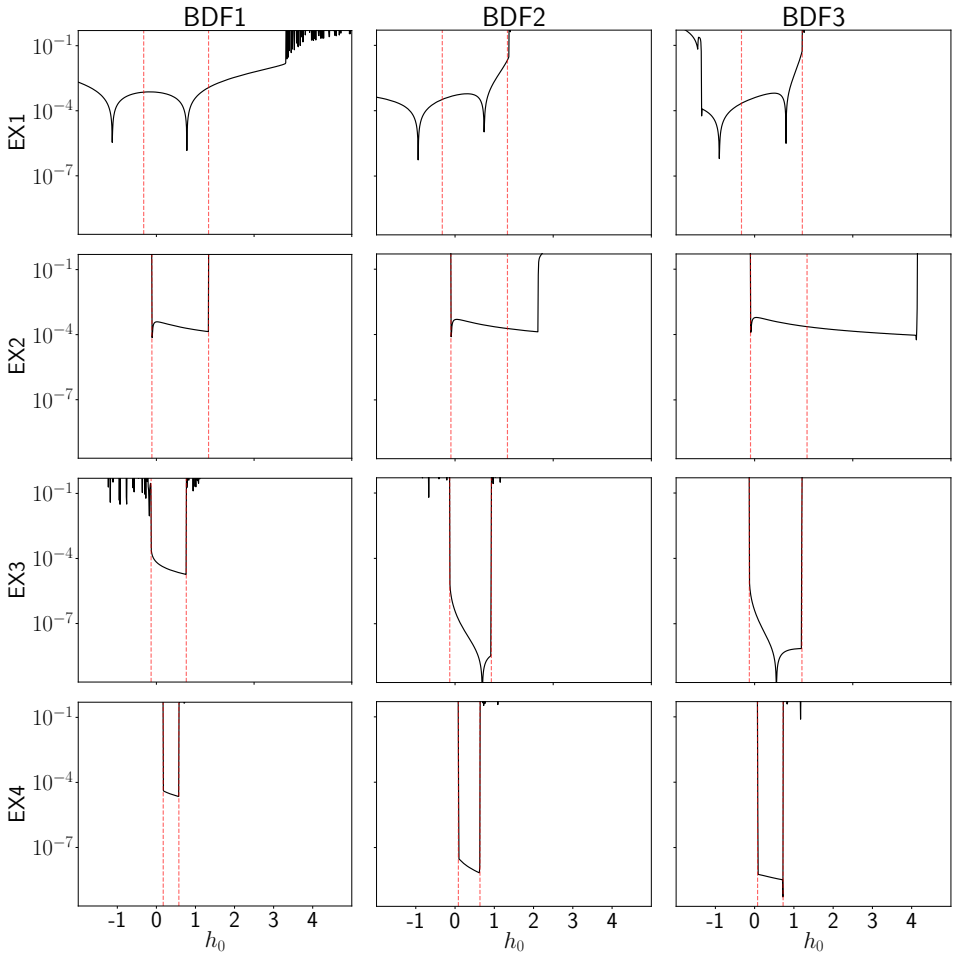


Figure 4.2: Error ($|T - T^{\text{ex}}| / T^{\text{ex}}$) at $t = 1$ for the test case in Section 4.4 as a function of the enthalpy offset h_0, using method #1. The red vertical dotted lines bound the values for which Eq. 4.25 held at all time steps.

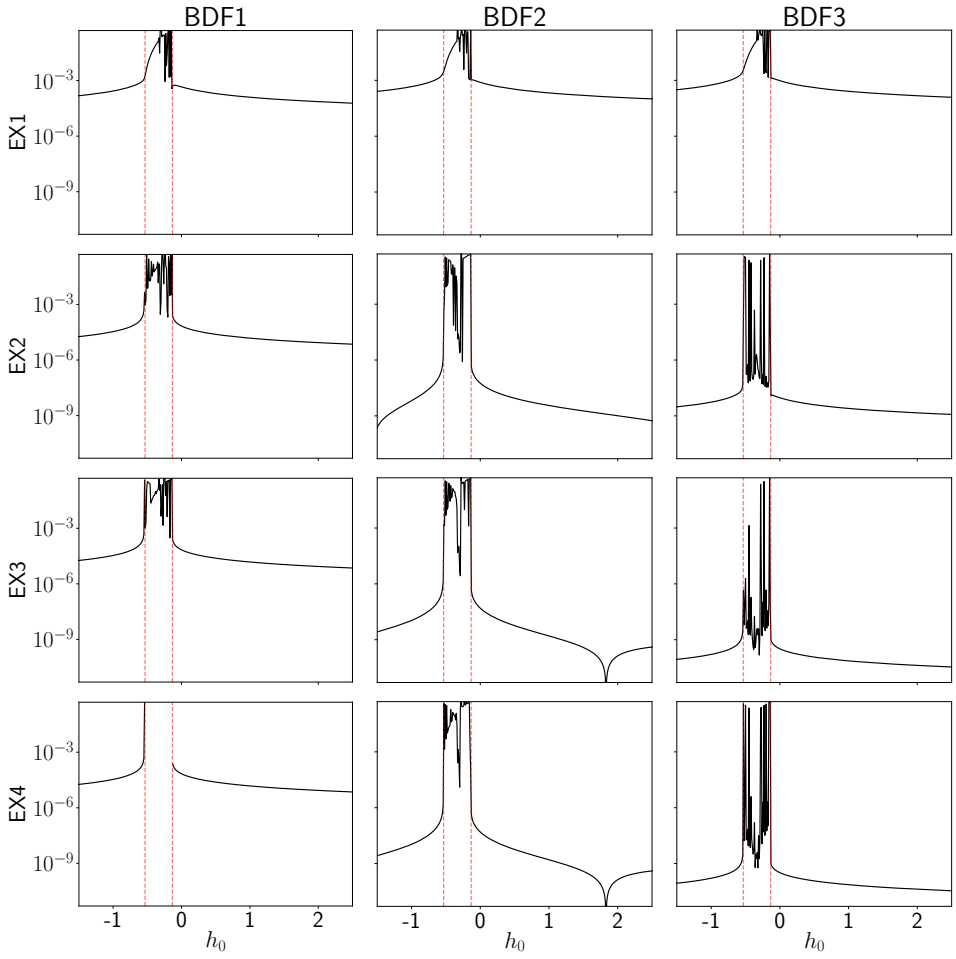


Figure 4.3: Error ($|T - T^{\mathrm{ex}}| / T^{\mathrm{ex}}$) at $t = 1$ for the test case in Section 4.4 as a function of the enthalpy offset h_0, using method #2. The red vertical dotted lines bound the values for which the stability criterion in Eq. 4.40 is violated at some time t.

The exact solution is

$$\mathbf{m}^{\text{ex}} = \frac{1}{4}\left(1+t^3\right)\begin{bmatrix}(x/L-1)^3\\(x/L-1)^2\,(y-1)(y+1)\end{bmatrix}+\begin{bmatrix}3/2\\0\end{bmatrix}, \qquad (4.41)$$
$$p^{\text{ex}} = \left(1+t^3\right)(L-x)^3$$

with $0 < t \leq 1$, and the domain is as in section 3.3.2, Fig. 3.2. The addition of $[3/2, 0]^\intercal$ ensures that $m_1^{\text{ex}} > 0$, so that there is no outflow at the Dirichlet boundary condition, and no inflow at the outlet. The density is determined by integrating $(-\nabla \cdot \mathbf{m})$ over t, to find

$$\rho^{\text{ex}} = -\frac{1}{4}\left(\frac{1}{4}t^4 + t\right)\,(x/L-1)^2(2y+3/L)\;+\;3\,, \qquad (4.42)$$

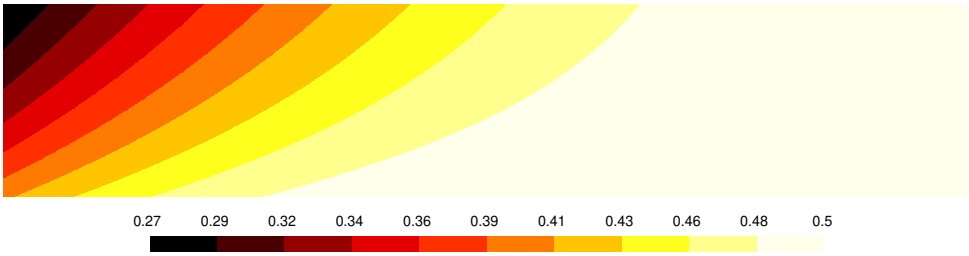
where the addition of the constant 3 ensures that $\rho > 0$ everywhere. The specific heat capacity is constant, so that $h = c_p T - h_0$ with $c_p = 1$. We use a non-affine equation of state: $T = ((\rho_1 - \rho)/(\rho_1 - \rho_0))^2$, where $\rho_0 = 2$ and $\rho_1 = 4$ are lower and upper bounds for ρ, so that the temperature is between 0 and 1. As in Section 3.3.2, the viscosity and conductivity are $\mu = 0.1 + T(1-T)$ and $k = \mu c_p/\text{Pr}$ with $\text{Pr} = 1$. The solution is depicted in Fig. 4.4.

We base the enthalpy time-stepping scheme on a linearization of $(\rho h)^n$ about a predictor for h^n, that is, method #2 in Section 4.3. We let $h_0 = 0.2$, so that Eq. 4.28 is satisfied everywhere. Increasing h_0 had no noticeable effect. The predictor h^* is obtained with a second-order extrapolation from previous time steps (using the EX2 weights in Table 4.1). We found that increasing the extrapolation order for h^* had no noticeable effect on the stability or the errors, which is in line with the tests for the BDF2 scheme in Fig. 4.3.

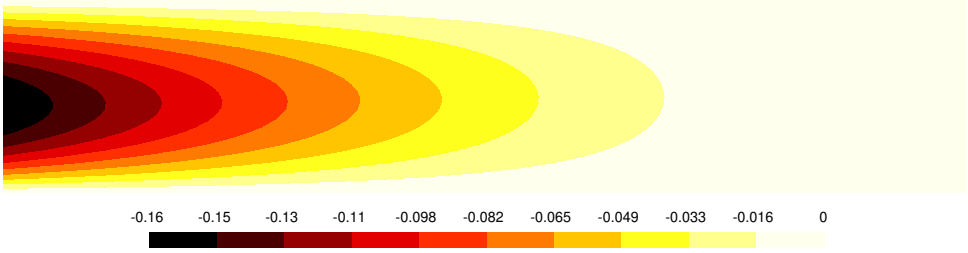
Fig. 4.5 shows the convergence with temporal refinement. The velocity and the temperature converge with second order, just like for the constant-density results in Fig 3.4, though in this case the pressure also shows $\mathcal{O}\left(\delta t^2\right)$ behavior. Note that the mixed-order cases remain fully stable, even for very small δt, despite the possible small-δt instability for variable-density flows that was discussed in Section 4.3.

Fig. 4.6 shows two other examples of temporal convergence with a mixed-order scheme, but with less effective enthalpy treatments. We found that method #1 was stable for all $h_0 \geq 0$, and that the precise value of h_0 is of little consequence to the L2 errors in the final answer. It is clear that the temperature does not converge with second order when method #1 is used for the enthalpy treatment. The right column in Fig. 4.6 shows that method #2 can also be unstable when the enthalpy offset is not sufficiently large (here it is 0.0 instead of 0.2): all quantities still converge with second-order accuracy, but the error diverges at small time steps. Some calculations failed due to numerical backflow at the outlet, which our simple outlet boundary condition cannot handle.

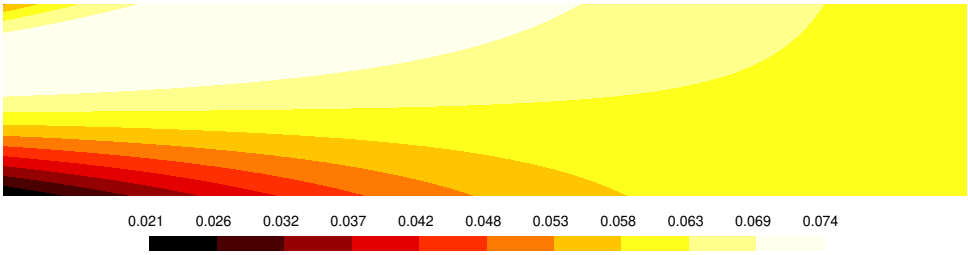
Table 4.3 collects the spatial convergence rates that were obtained with method #2 and $h_0 = 0.2$ (i.e., the same conditions are for the temporal convergence in Fig. 4.5). As the mesh is refined, the mixed-order discretization displays $\mathcal{O}\left(\ell^{\mathcal{P}+1}\right)$ behavior for quantities with a polynomial order $\mathcal{P}$, where $\ell \propto 1/N_y$ is the characteristic



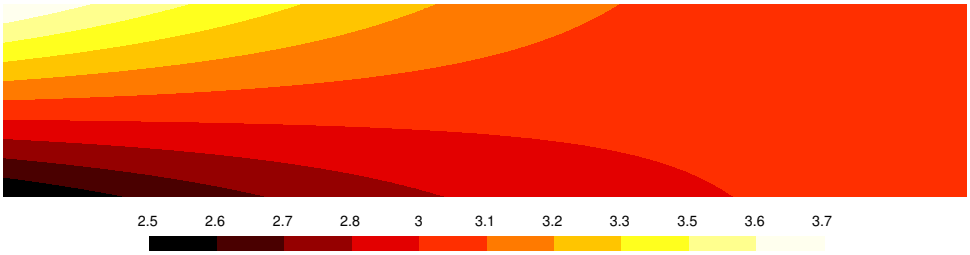
(a) u_1



(b) u_2



(c) Kinematic viscosity (ν) and thermal diffusivity (α). (Note that $\mathrm{Pr} = 1$, so $\alpha = \nu$.)



(d) ρ

Figure 4.4: Variable-density manufactured solution in Eqs. 4.41-4.42 at $t = 1$.

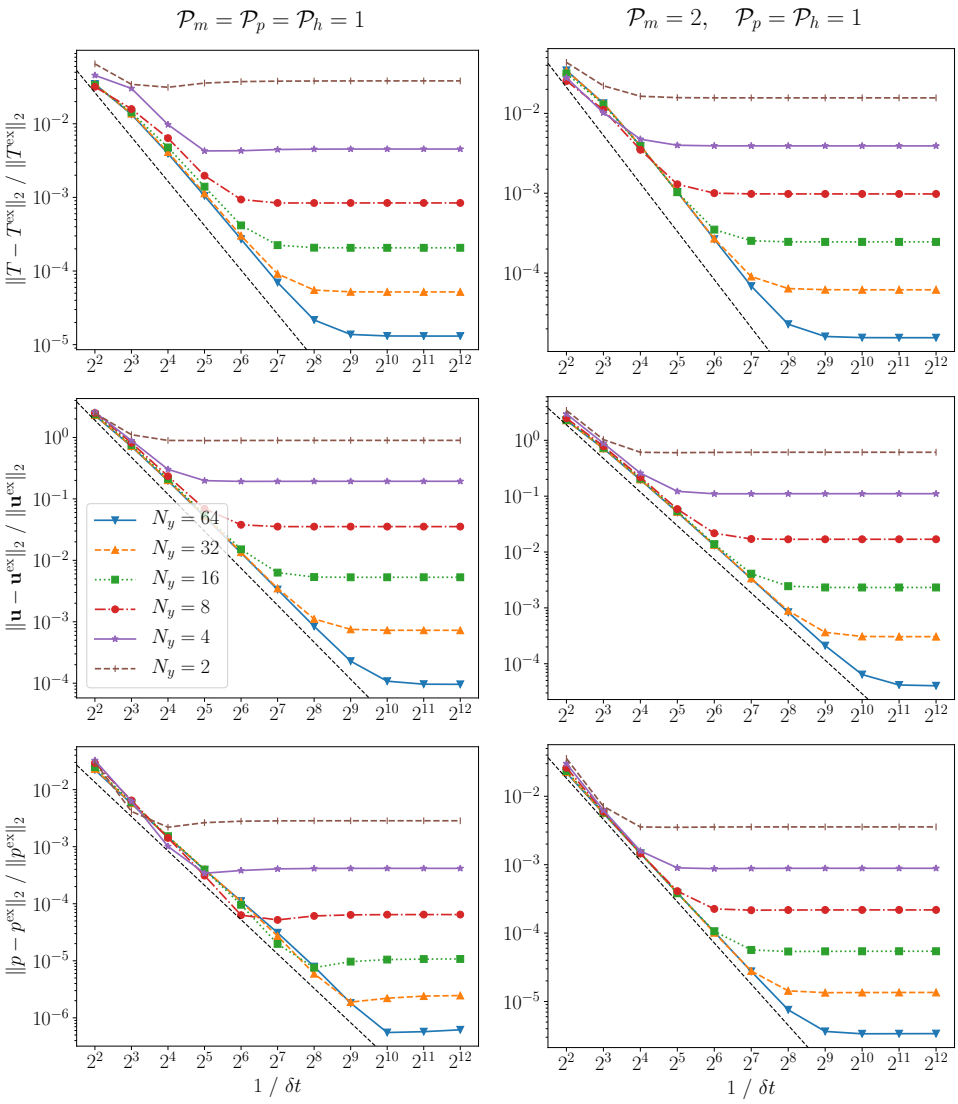


Figure 4.5: Convergence of the numerical solution toward the variable-density manufactured solution (Eq. 4.41) with temporal refinement, using method #2 with $h_0 = 0.2$. The characteristic element length is inversely proportional to N_y. The black dashed lines indicate ideal second-order convergence in δt.

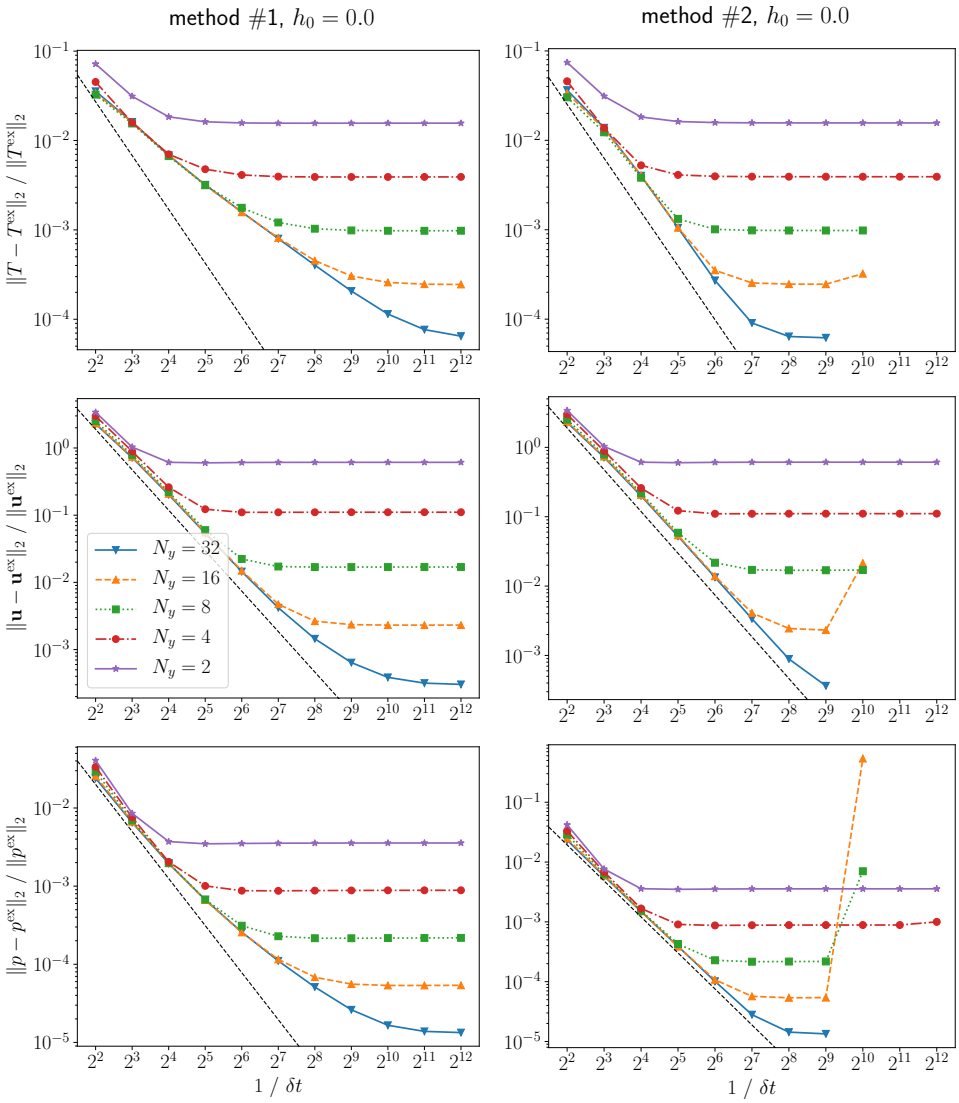


Figure 4.6: Equivalent of the mixed-order case ($\mathcal{P}_m = 2$, $\mathcal{P}_p = \mathcal{P}_h = 1$) in Fig. 4.5, but with a different value of h_0, and comparing method #1 to method #2. The missing values in the figures in the right column indicate failed calculations.

Table 4.3: Convergence toward the variable-density manufactured solution in Eqs. 4.41–4.42 (Fig. 4.4) with spatial refinement and fixed $\delta t = 2^{-12}$.

	temperature		velocity		pressure	
N_y	error	conv	error	conv	error	conv
Mixed order $(\mathcal{P}_m = 2,\ \mathcal{P}_h = \mathcal{P}_p = 1)$*:*						
2^1	1.56e-2		6.11e-1		3.57e-3	
2^2	3.92e-3	2.00	1.11e-1	2.47	8.86e-4	2.01
2^3	9.80e-4	2.00	1.69e-2	2.71	2.18e-4	2.02
2^4	2.46e-4	2.00	2.32e-3	2.87	5.43e-5	2.01
2^5	6.17e-5	1.99	3.05e-4	2.92	1.36e-5	2.00
2^6	1.55e-5	1.99	4.03e-5	2.92	3.39e-6	2.00
Equal order $(\mathcal{P}_m = \mathcal{P}_p = \mathcal{P}_h = 1)$*:*						
2^1	3.82e-2		8.96e-1		2.85e-3	
2^2	4.54e-3	3.07	1.93e-1	2.21	4.17e-4	2.01
2^3	8.41e-4	2.43	3.54e-2	2.45	6.46e-5	2.02
2^4	2.07e-4	2.02	5.30e-3	2.74	1.08e-5	2.01
2^5	5.20e-5	1.99	7.26e-4	2.87	2.46e-6	2.00
2^6	1.31e-5	1.99	9.59e-5	2.92	6.17e-7	2.00

mesh length. For the equal-order case, the velocity shows $\mathcal{O}\left(\ell^{\mathcal{P}_u+2}\right)$ hyperconvergence in $\mathbf{u}$.

Paradoxically, the errors in the pressure are much lower for the equal-order case, despite its a reduced solution space. This could perhaps be explained by the error in the $\partial\rho/\partial t$ term, which is extrapolated from previous time steps (Eq. 4.12). This might induce discontinuities in p, which would be suppressed by the pressure stabilization, which is only present for the equal-order discretization. This explanation is supported by the fact that this phenomenon did not occur for the constant-density manufactured solution in section 3.3.2 (Table 3.3). Note that the orders of convergence are the same for constant-density and the variable-density test cases.

4.5.2. Validation with Flow Past a Heated Circular Obstacle

To validate the numerical method with a variable-density flow, we replicate a numerical test case by Shi *et al.* [29], who used a specialized cylindrical finite volume scheme to handle the circular geometry. The geometry is the same as in section 3.4, see Figs. 3.5. The temperature is fixed at T_w at the cylinder, and T_∞ at the inlet, top and bottom parts of the domain. The outlet has a homogeneous Neumann boundary condition. The temperatures are $T_\infty = 20°\mathrm{C}$ and $T_\mathrm{w} = 1.5T_\infty = 166.575°\mathrm{C}$, resulting in $\mathrm{Pr} = 0.7146$. Shi *et al.* solved for the temperature, approximating the material properties as

$$\varphi = a_0 + a_1(T - T_F) + a_2(T - T_F)^2 \ , \tag{4.43}$$

Table 4.4: Coefficients for the material properties in Eq. 4.43. Reproduced from [29].

	ρ (kg m^{-3})	μ (kg $m^{-1}s^{-1}$)	k (W $m^{-1}K^{-1}$)	c_p ($m^2s^{-2}K^{-1}$)
a_0	1.268672727	1.7254e-05	2.4195e-2	1.00620979e3
a_1 (K^{-1})	-4.08741e-03	4.95611e-08	7.5234e-5	1.4522145e-2
a_2 (K^{-2})	7.23864e-06	-2.7214e-11	-3.2588e-8	4.13753e-4

where φ is one of (ρ, μ, k, c_p), $T_F = 0°\mathrm{C}$, and the coefficients a_i are in Table 4.4. Since $c_p := h_T$ is a second-order polynomial in T, we need to find the root of the third-order polynomial $h = h(T)$ to map from h to a fluid property. This minor inconvenience permits a better comparison with the results in Shi *et al.* .

We obtain our results on the same mesh as for the isothermal case (Fig. 3.6). Fig. 4.7 shows an example of instantaneous flow fields. Fig. 4.8 shows the lift and drag coefficients and the Nusselt number, which is defined as

$$\mathrm{Nu} = \frac{D}{T_\mathrm{w} - T_\infty} \frac{1}{\|\partial S\|_\mathrm{leb}} \int_{\partial S} \mathbf{n} \cdot \nabla T \ , \tag{4.44}$$

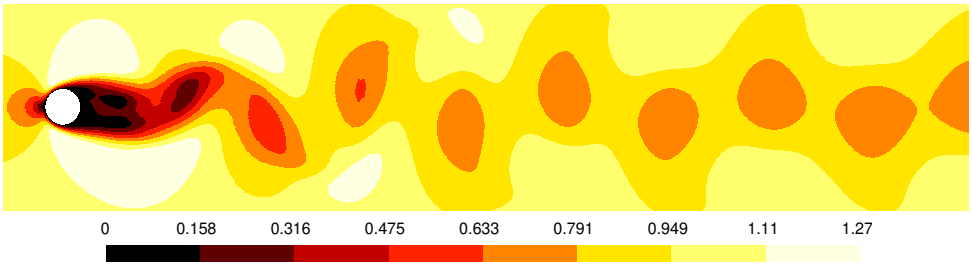
where $\|\partial S\|_\mathrm{leb} = \pi D$ is the circumference of the circular obstacle. Recall that the dimensionless shedding frequency was approximately $\mathrm{St} = 0.166$ for the isothermal case in section 3.4. Here we find $\mathrm{St} = 0.1536$ for the heated cylinder, which differs by 1% from the value of $\mathrm{St} = 0.152$ in Shi *et al.* [29] and the experimental value of $\mathrm{St} = 0.152$ in [30].

This result was obtained by linearizing $(\rho h)^n$ with method #2 for the temporal derivative of the enthalpy. We subtracted an offset h_0 from the enthalpy (as explained in Section 4.3.2), such that the maximum value of h was zero. Interestingly, there is no noticeable change in the results when we use an equal-order scheme, even when δt is decreased by a factor of 10 or 100. For the mixed-order calculations, we found no difference between using an SIP or an LDG pressure matrix. We repeated the calculation using method #1, setting the enthalpy offset to a value h_0^*, such that $h = 0$ at $T = (T_\mathrm{w} + T_\infty)/2$. This resulted in almost exactly the same shedding frequency ($\mathrm{St} = 0.1537$).

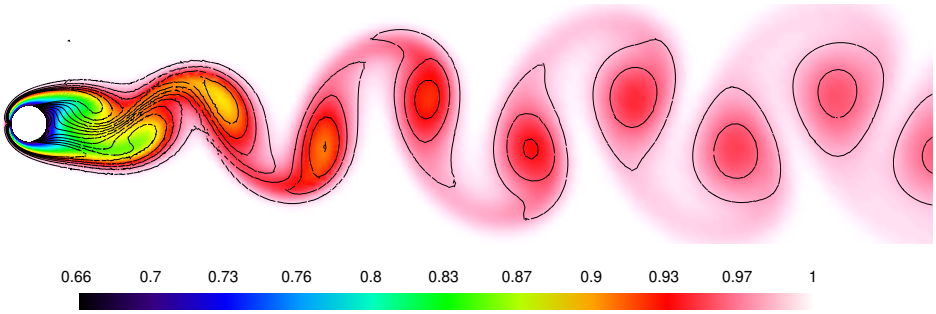
For all test cases, we found that some values for the enthalpy offset result in unstable schemes, yielding oscillatory pressure fields. For method #1 this happens when h_0 is far from h_0^*; for method #2 this happens when h_0 is too small. When operating in the range of stable h_0 values, the exact enthalpy offset has no noticeable impact on the shedding frequency.

4.6. Discussion and Conclusion

Since the density is a function of the specific enthalpy, the temporal finite difference scheme requires that the volumetric enthalpy (ρh) be linearized in h, and we have analyzed two methods for doing this, both of which need a predictor for the enthalpy at the new time step. This led to theoretical stability requirements in case the enthalpy equation is iterated within a time step, or, equivalently, in the limit of



(a) $|m|/(\rho u)_\infty$



(b) Colored field of ρ/ρ_∞, overlaid with isolines of $\mathrm{curl}(u)D/u_\infty$.

Figure 4.7: Instantaneous fields for flow past a heated circular cylinder.

small time steps. The specific enthalpy is shifted with an offset h_0, so that solving for the new unknown $h - h_0$ satisfies these stability requirements. These results were verified with simple space-independent tests in Section 4.4.

Method #1 is basically what has been done in all previous literature we have seen (e.g., [20], [18], [15]), but for this approach we cannot determine the range of stable h_0 values a priori. Another disadvantage is that full accuracy with a second-order time-stepping scheme (BDF2) can only be achieved if the predictor is third-order accurate. This necessitates storing three instead of two previous time steps for the enthalpy, and the third-order extrapolation negatively affects the stability (see Fig. 4.2).

The other linearization of (ρh) (method #2) does provide full accuracy when the predictor is extrapolated from two previous time steps, in which case the error in the linearization is negligible in the limit of small time steps. Furthermore, the stability requirement is simply that the volumetric enthalpy be a monotonic function of the temperature (or, equivalently, the specific enthalpy), and this leads to a range of stable h_0 values that can be determined a priori. This results in a stable scheme, in which the error of the linearization of (ρh) becomes negligible in the limit of small time steps. The manufactured solutions demonstrate full second-order temporal accuracy, without any predictor steps.

In a real flow simulation the exact range of stable values for h_0 cannot be determined a priori due the coupling of the transport equations. Note that a discon-

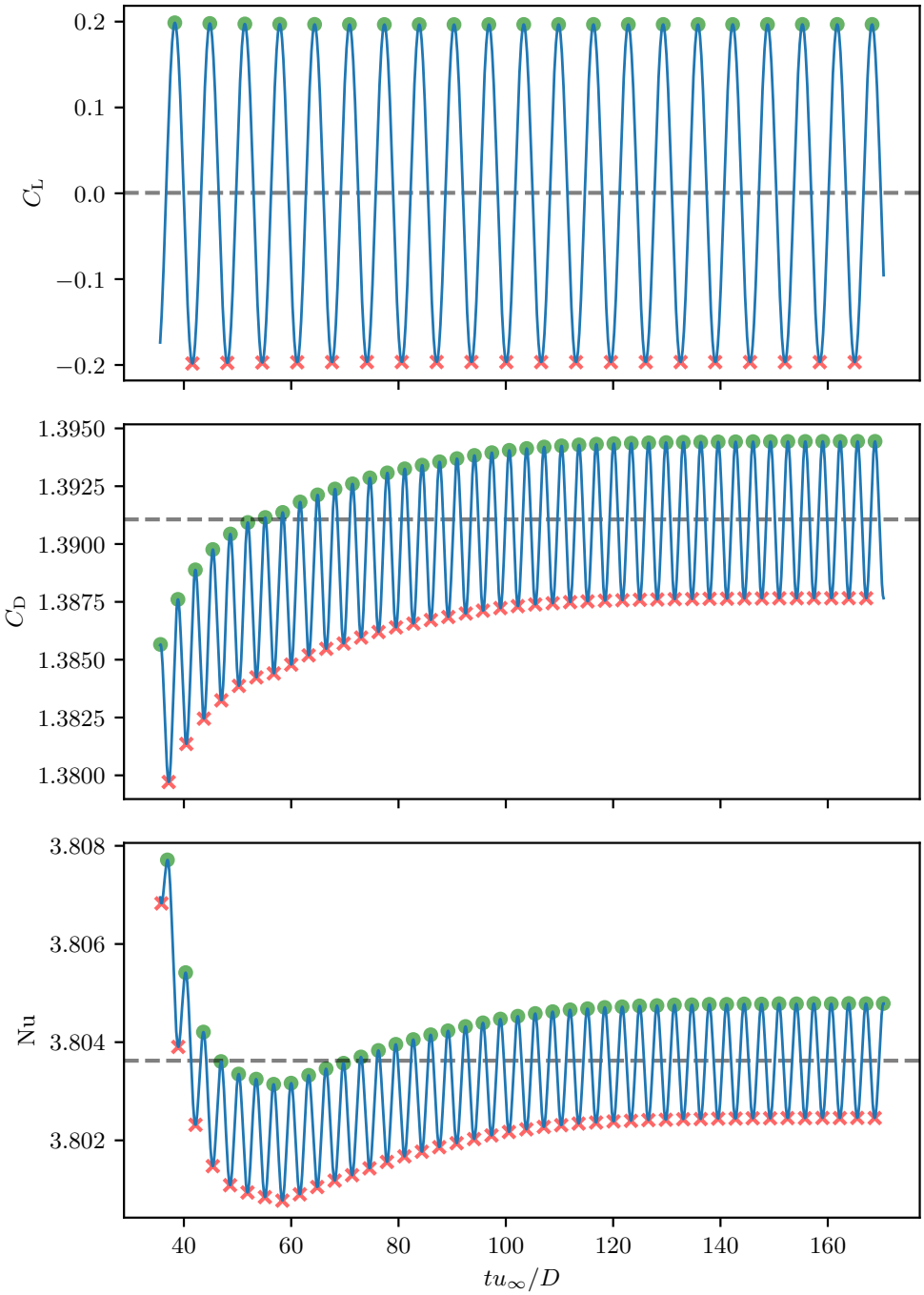


Figure 4.8: Temporal behavior of the lift and drag coefficients (Eqs. 3.18), and Nusselt number (Eq. 4.44) for flow past a heated circular cylinder. The crosses and circles mark local minima and maxima. The dashed horizontal lines indicate the averages over the last five periods.

tinuous Galerkin discretizations does not guarantee that the extreme values of the numerical solution lie within physically acceptable bounds, so that the numerical temperature range is not known beforehand. Nevertheless, there are useful guidelines: for method #1, $h - h_0$ should be close to zero; for method #2, h_0 should be sufficiently large. Once a stable h_0 value is found, our numerical experiments in Section 4.5 suggest that the exact value of h_0 has little bearing on the overall accuracy.

Comparing the present manufactured solution to the one from the previous chapter suggests that the temporal finite difference approximation for $\partial\rho/\partial t$ (Eq. 4.12) is a significant source of error (as explained at the end of section 4.5.1), which would be in line with previous literature. An elegant improvement could be to approximate $\partial\rho/\partial t$ by the spatial discretization of $\partial(\rho h)/\partial t$, as mentioned in section 4.2.

We are now finally ready to look at the bigger picture, and reconsider which combination of transport equation (primitive or conservative) and unknown (h or H) is most suitable for discretization in low-Mach number flows. Table 4.5 summarizes the findings in this chapter. Solving for the volumetric enthalpy would also have been a reasonable option. Given how much depends on the enthalpy offset, it is surprising that we have not seen this notion mentioned in previous literature.

A. Derivations of the Results in Section 4.3.1

The error estimates that were presented in Section 4.3.1 are derived here. To ease the notation, let

$$K \coloneqq \rho_h^* \quad \text{and} \quad B \coloneqq \frac{1}{2}\rho_{hh}^* \,, \tag{4.45}$$

so that the Taylor expansion in Eq. 4.20 becomes

$$\rho^n = \rho^* - K\epsilon^* + B{\epsilon^*}^2 + \mathcal{O}\left({\epsilon^*}^3\right) \,. \tag{4.46}$$

A.1. Derivations for Method #1

To derive Eq. 4.24, subtract Eq. 4.16 from Eq. 4.17 to get

$$\rho^* h^{[1]} - \rho^n h^n = -\frac{\delta t}{\gamma_0}\lambda\ \epsilon^{[1]} \,. \tag{4.47}$$

Substituting Eq. 4.46 gives

$$\rho^* \epsilon^{[1]} + \left(K\epsilon^* + \mathcal{O}\left({\epsilon^*}^2\right)\right) h^n = -\frac{\delta t}{\gamma_0}\lambda\ \epsilon^{[1]} \,, \tag{4.48}$$

which can be rearranged to

$$\frac{\epsilon^{[1]}}{\epsilon^*} = \left(\frac{-K}{\rho + \lambda\,(\delta t/\gamma_0)}\right)^* h^n + \mathcal{O}\left(\epsilon^*\right) \,. \tag{4.49}$$

Since $h^n = h^* - \epsilon^*$, this is indeed equivalent to Eq. 4.24.

Table 4.5: Overview of problems that must be overcome in various areas, depending on the form of the transport equation (primitive or conservative), and the choice of the unknown variable (primitive h or conserved $H \coloneqq \rho h$). A blank cell indicates that the treatment is straightforward.

	conservative equation		primitive equation
	solve for H	solve for h	solve for h
temporal term $\mathrm{d}/\mathrm{d}t$	-	can be unconditionally stable with a negligible error*	-
advection term	-	-	unclear, non-trivial with FEM
diffusion term	needs modification**	-	extra term**: $\alpha \mathbf{d} \cdot \nabla h$
mapping from unknown to fluid properties	requires special care†	-	-
replacing $\partial\rho/\partial t$ in continuity equation by enthalpy discretization‡	depends on enthalpy offset h_0 (untested)	depends on enthalpy offset h_0 (untested)	-

* As shown in sections 4.3 and 4.4, using method #2, and depending on the enthalpy offset h_0.
** See the discussion in section 4.1.2.
† Can be done with a suitable enthalpy offset h_0 (ref. Fig. 4.1), or with a separate density predictor, as discussed in section 4.1.2.
‡ As suggested by Irrenfried [22, 23], see section 4.2.

This result can now be used to derive the error estimates in Eq. 4.23a. Using the fact that $h^{[1]} = h^n + \epsilon^{[1]}$, we can write

$$\rho^* h^{[1]} = \rho^n h^n - (\rho^n - \rho^*)\, h^n + \rho^* \epsilon^{[1]} \,, \tag{4.50}$$

which, upon substitution into Eq. 4.17, gives

$$\begin{aligned} &\frac{\gamma_0}{\delta t}(\rho h)^n + \sum_{i=1}^{q} \frac{\gamma_i}{\delta t}(\rho h)^{n-i} \\ &= -\left(\lambda - \frac{\gamma_0}{\delta t}(\rho^n - \rho^*)\right) h^n + Q^n - \frac{\gamma_0}{\delta t}\left(\rho^* + \frac{\delta t}{\gamma_0}\lambda\right)\epsilon^{[1]} \,. \end{aligned} \tag{4.51}$$

4

Using the error estimate in Eq. 4.49,

$$\begin{aligned} &\frac{\gamma_0}{\delta t}(\rho h)^n + \sum_{i=1}^{q} \frac{\gamma_i}{\delta t}(\rho h)^{n-i} \\ &= -\left(\lambda - \frac{\gamma_0}{\delta t}\left(\rho^n - \rho^* + \rho_h^* \epsilon^*\right)\right) h^n + Q^n + \mathcal{O}\left({\epsilon^*}^2/\delta t\right) \\ &= -\left(\lambda + \mathcal{O}\left({\epsilon^*}^2/\delta t\right)\right) h^n + Q^n + \mathcal{O}\left({\epsilon^*}^2/\delta t\right) \,, \end{aligned} \tag{4.52}$$

where the second equality follows from the Taylor expansion in Eq. 4.20.

A.2. Derivations for Method #2

In analogy with the previous subsection, Eq. 4.26 is derived by subtracting Eq. 4.16 from Eq. 4.19 to get

$$(\rho h)_h^* h^{[2]} - \rho^n h^n = \left(h^2 \rho_h\right)^* - \frac{\delta t}{\gamma_0}\lambda\, \epsilon^{[2]} \,. \tag{4.53}$$

Upon substituting Eq. 4.46, this becomes

$$(\rho h)_h^* h^{[2]} - h^n \rho^* + K h^n \epsilon^* - B h^n {\epsilon^*}^2 + \mathcal{O}\left({\epsilon^*}^3\right) = \left(h^2 \rho_h\right)^* - \frac{\delta t}{\gamma_0}\lambda\, \epsilon^{[2]} \,. \tag{4.54}$$

Substituting $(\rho h)_h = \rho + K h$ on the left-hand side, and $\left(h^2 \rho_h\right)^* = K h^* (h^n + \epsilon^*)$ on the right-hand side, gives

$$(\rho + K h)^* h^{[2]} - h^n \rho^* + K h^n \epsilon^* - B h^n {\epsilon^*}^2 + \mathcal{O}\left({\epsilon^*}^3\right) = K h^* (h^n + \epsilon^*) - \frac{\delta t}{\gamma_0}\lambda\, \epsilon^{[2]} \,. \tag{4.55}$$

This can be rearranged to

$$\rho^*(h^{[2]} - h^n) + K h^*(h^{[2]} - h^n) - K(h^* - h^n)\epsilon^* - B h^n {\epsilon^*}^2 + \frac{\delta t}{\gamma_0}\lambda\, \epsilon^{[2]} = \mathcal{O}\left({\epsilon^*}^3\right) \,. \tag{4.56}$$

Using the definitions for $\epsilon^{[2]}$ and ϵ^*, this becomes

$$\left(\rho + Kh + \frac{\delta t}{\gamma_0}\lambda\right)^* \epsilon^{[2]} - \left(K + Bh^n\right)\epsilon^{*2} = \mathcal{O}\left(\epsilon^{*3}\right) . \tag{4.57}$$

Recalling $h^n = h^* - \epsilon^*$, this is indeed equivalent to Eq. 4.26.

This result can now be used to derive the error estimates in Eq. 4.23b. Upon substituting

$$(\rho h)_h^* h^{[2]} = \rho^* h^{[2]} + Kh^* h^{[2]} = \rho^n h^n - (\rho^n - \rho^*)\, h^n + \rho^* \epsilon^{[2]} + Kh^* h^{[2]} , \tag{4.58}$$

Eq. 4.19 becomes

$$\begin{aligned} &\frac{\gamma_0}{\delta t}\left(\rho^n h^n - (\rho^n - \rho^*)\, h^n + \rho^* \epsilon^{[2]} + Kh^* h^{[2]}\right) + \sum_{i=1}^{q} \frac{\gamma_i}{\delta t} (\rho h)^{n-i} \\ =&\frac{\gamma_0}{\delta t} K h^{*2} - \lambda h^{[2]} + Q^n \\ =&\frac{\gamma_0}{\delta t} K h^* \left(h^{[2]} - \left(\epsilon^{[2]} - \epsilon^*\right)\right) - \lambda\left(h^n + \epsilon^{[2]}\right) + Q^n , \end{aligned} \tag{4.59}$$

which can be rearranged to

$$\begin{aligned} &\frac{\gamma_0}{\delta t} (\rho h)^n + \sum_{i=1}^{q} \frac{\gamma_i}{\delta t} (\rho h)^{n-i} \\ =&\frac{\gamma_0}{\delta t}\left((\rho^n - \rho^*)\, h^n - \rho^* \epsilon^{[2]} + Kh^*\left(\epsilon^* - \epsilon^{[2]}\right) - \frac{\delta t}{\gamma_0}\lambda \epsilon^{[2]}\right) - \lambda h^n + Q^n . \end{aligned} \tag{4.60}$$

Eq. 4.57 can now be used to eliminate $\epsilon^{[2]}$:

$$\begin{aligned} &\frac{\gamma_0}{\delta t} (\rho h)^n + \sum_{i=1}^{q} \frac{\gamma_i}{\delta t} (\rho h)^{n-i} \\ =&\frac{\gamma_0}{\delta t}\left((\rho^n - \rho^*)\, h^n + Kh^* \epsilon^* - \left(K + Bh^n\right)\epsilon^{*2} + \mathcal{O}\left(\epsilon^{*3}\right)\right) - \lambda h^n + Q^n \\ =&\frac{\gamma_0}{\delta t}\left(\left(\rho^n - \rho^* - B\epsilon^{*2}\right) h^n + K\epsilon^* (h^* - \epsilon^*) + \mathcal{O}\left(\epsilon^{*3}\right)\right) - \lambda h^n + Q^n \\ =&\frac{\gamma_0}{\delta t}\left(\left(\rho^n - \rho^* - B\epsilon^{*2} + K\epsilon^*\right) h^n + \mathcal{O}\left(\epsilon^{*3}\right)\right) - \lambda h^n + Q^n . \end{aligned} \tag{4.61}$$

Finally, using the Taylor series in Eq. 4.46,

$$\frac{\gamma_0}{\delta t} (\rho h)^n + \sum_{i=1}^{q} \frac{\gamma_i}{\delta t} (\rho h)^{n-i} = \frac{\gamma_0}{\delta t}\left(\mathcal{O}\left(\epsilon^{*3}\right)\, h^n + \mathcal{O}\left(\epsilon^{*3}\right)\right) - \lambda h^n + Q^n , \tag{4.62}$$

which implies Eq. 4.22 with Eq. 4.23b.

References

[1] A. Hennink, M. Tiberga, and D. Lathouwers, *A Pressure-based solver for low-Mach number flow using a discontinuous Galerkin method,* Journal of Computational Physics **425**, 109877 (2021).

[2] G. Volpe, *Performance of compressible flow codes at low Mach numbers,* AIAA Journal **31**, 49 (1993).

[3] S. S. Collis, *Discontinuous Galerkin methods for turbulence simulation,* in *Summer Program 2002, Center for Turbulence Research* (2002) pp. 155–167.

[4] F. Föll, S. Pandey, X. Chu, C.-D. Munz, E. Laurien, and B. Weigand, *High-Fidelity Direct Numerical Simulation of Supercritical Channel Flow Using Discontinuous Galerkin Spectral Element Method,* in *High Performance Computing in Science and Engineering ' 18* (Springer International Publishing, Cham, 2019) pp. 275–289.

[5] R. Barney, R. Nourgaliev, J. P. Delplanque, and R. McCallen, *Fully-Implicit, High-Order, Reconstructed Discontinuous Galerkin Method for Supercritical Fluid Flows,* in *Proceedings of the International Conference on Mathematics and Computational Methods applied to Nuclear Science and Engineering (M&C)* (Portland, OR, USA, 2019) pp. 377–386.

[6] A. Nigro, C. De Bartolo, R. Hartmann, and F. Bassi, *Discontinuous Galerkin solution of preconditioned Euler equations for very low Mach number flows,* International Journal for Numerical Methods in Fluids **63**, 449 (2010).

[7] F. Bassi, A. Crivellini, D. A. D. Pietro, and S. Rebay, *An artificial compressibility flux for the discontinuous Galerkin solution of the incompressible Navier-Stokes equations,* Journal of Computational Physics **218**, 794 (2006).

[8] A. Crivellini, V. D'Alessandro, and F. Bassi, *A Spalart-Allmaras turbulence model implementation in a discontinuous Galerkin solver for incompressible flows,* Journal of Computational Physics **241**, 388 (2013).

[9] K. Shahbazi, P. F. Fischer, and C. R. Ethier, *A high-order discontinuous Galerkin method for the unsteady incompressible Navier-Stokes equations,* Journal of Computational Physics **222**, 391 (2007).

[10] L. Botti and D. A. Di Pietro, *A pressure-correction scheme for convection-dominated incompressible flows with discontinuous velocity and continuous pressure,* Journal of Computational Physics **230**, 572 (2011).

[11] S. Rhebergen, B. Cockburn, and J. J. W. van der Vegt, *A space-time discontinuous Galerkin method for the incompressible Navier-Stokes equations,* Journal of Computational Physics **233**, 339 (2013).

[12] M. Piatkowski, S. Müthing, and P. Bastian, *A stable and high-order accurate discontinuous Galerkin based splitting method for the incompressible Navier-Stokes equations,* Journal of Computational Physics **356**, 220 (2018).

[13] F. Bassi, L. Botti, A. Colombo, A. Ghidoni, and F. Massa, *Linearly implicit Rosenbrock-type Runge-Kutta schemes applied to the Discontinuous Galerkin solution of compressible and incompressible unsteady flows,* Computers & Fluids **118**, 305 (2015).

[14] L. Pesch and J. van der Vegt, *A discontinuous Galerkin finite element discretization of the Euler equations for compressible and incompressible fluids,* Journal of Computational Physics **227**, 5426 (2008).

[15] B. Klein, B. Müller, F. Kummer, and M. Oberlack, *A high-order discontinuous Galerkin solver for low Mach number flows,* International Journal for Numerical Methods in Fluids **81**, 489 (2016).

[16] B. Klein, *A high-order Discontinuous Galerkin solver for incompressible and low-Mach number flows*, Ph.D. thesis, Technische Universität, Darmstadt (2015).

[17] D. T. Banuti, *Crossing the Widom-line – Supercritical pseudo-boiling,* The Journal of Supercritical Fluids **98**, 12 (2015).

[18] J. W. R. Peeters, *Turbulence and turbulent heat transfer at supercritical pressure*, doctoral thesis, Delft University of Technology (2016).

[19] H. N. Najm, P. S. Wyckoff, and O. M. Knio, *A Semi-implicit Numerical Scheme for Reacting Flow: I. Stiff Chemistry,* Journal of Computational Physics **143**, 381 (1998).

[20] H. Nemati, *Direct numerical simulation of turbulent heat transfer to fluids at supercritical pressures*, doctoral thesis, TU Delft (2016).

[21] F. Nicoud, *Conservative High-Order Finite-Difference Schemes for Low-Mach Number Flows,* Journal of Computational Physics **158**, 71 (2000).

[22] C. Irrenfried, *DNS and experimentally based modelling of convective turbulent near wall heat transfer at high Prandtl numbers*, Doctoral thesis, Graz University of Technology, Fakultät für Maschinenbau und Wirtschaftswissenschaften, Institut für Strömungslehre und Wärmeübertragung (2019).

[23] C. Irrenfried, *Convective turbulent near wall heat transfer at high Prandtl numbers. Monograph Series: Computation in Engineering and Science*, Vol. 37 (Technischen Universität Graz, 2020).

[24] R. Span and W. Wagner, *A New Equation of State for Carbon Dioxide Covering the Fluid Region from the Triple-Point Temperature to 1100 K at Pressures up to 800 MPa,* Journal of Physical and Chemical Reference Data **25**, 1509 (1996).

[25] I. H. Bell, J. Wronski, S. Quoilin, and V. Lemort, *Pure and Pseudo-pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library CoolProp,* Industrial & Engineering Chemistry Research **53**, 2498 (2014).

[26] A. Hennink, *Finite Difference Methods for the Non-linear Enthalpy Equation,* GitHub. [Online] Accessed on 2019-11-05 (2019).

[27] L. Shunn, F. Ham, and P. Moin, *Verification of variable-density flow solvers using manufactured solutions,* Journal of Computational Physics **231**, 3801 (2012).

[28] K. Salari and P. Knupp, *Code Verification by the Method of Manufactured Solutions*, Tech. Rep. (Sandia National Labs., Albuquerque, NM (US), Livermore, CA (US), 2000).

[29] J.-M. Shi, D. Gerlach, M. Breuer, G. Biswas, and F. Durst, *Heating effect on steady and unsteady horizontal laminar flow of air past a circular cylinder,* Physics of Fluids **16**, 4331 (2004).

[30] A.-B. Wang, Z. Trávníček, and K.-C. Chia, *On the relationship of effective Reynolds number and Strouhal number for the laminar vortex shedding of a heated circular cylinder,* Physics of Fluids **12**, 1401 (2000).

5

Channel Flow and Large Eddy Simulation

5.1. Introduction and Governing Equations

The idea of a large eddy simulation (LES) is to take spatially filtered quantities as the unknowns, in which the smallest scales of the flow have been removed. These filtered quantities are denoted by an overline (e.g., $\bar{p}$, $\overline{m_1}$, …). On a conceptual level, a filtered quantity $\overline{\phi}$ can be thought of as a moving average of ϕ, or a more general convolution of ϕ with a low-pass filter, though this is not always explicitly computed. The goal is to approximate the filtered quantities by solving some form of the transport equations in less detail, not down to the smallest length scales.

We solve for the variables $\overline{\mathbf{m}}$, $\bar{p}$, and $\widetilde{h} \coloneqq \overline{\rho h}/\bar{\rho}$, which are therefore known as the resolved quantities. For the specific enthalpy we use a Favre average (i.e., weighed by the density before filtering), denoted by a tilde. This is more suitable for primitive (non-conserved) variables. The fluid properties ($\widehat{\rho}$, $\widehat{\mu}$, $\widehat{\alpha}$) are determined from $\widetilde{h}$. A hat denotes a computable variable, meaning that it is based on the resolved quantities. The computable velocity is $\widehat{\mathbf{u}} = \overline{\mathbf{m}}/\widehat{\rho}$, which forms the basis for the resolved rate of strain

$$\widehat{S}_{ij} = L\left(\nabla\widehat{\mathbf{u}}\right) \ , \tag{5.1}$$

where the operator

$$L\left(A\right) \coloneqq \frac{1}{2}\left(A + A^{\mathsf{T}} - \frac{2}{3}\mathrm{trace}\left(A\right) I\right) \tag{5.2}$$

takes the symmetric, deviatoric part of a matrix in three dimensions. The resolved Fourier heat flux is $\widehat{\mathbf{q}} = -\widehat{\left(k/c_p\right)}\nabla\widetilde{h} = -\widehat{\rho\alpha}\ \nabla\widetilde{h}$.

Filtering does not commute with multiplication, and the commutation errors arise as extra terms in the transport equations. In particular, for the convective

term in the momentum equation, the difference between what we can compute and the filtered original term is

$$\widehat{\mathbf{u}}\overline{\mathbf{m}} - \overline{\mathbf{u}\mathbf{m}} = \frac{1}{\widehat{\rho}}\,\overline{\mathbf{m}}\overline{\mathbf{m}} - \overline{\rho\mathbf{u}}\overline{\mathbf{u}} = \frac{1}{\widehat{\rho}}\,\overline{\mathbf{m}}\overline{\mathbf{m}} - \bar{\rho}\widetilde{\mathbf{u}}\widetilde{\mathbf{u}} = \left(\frac{1}{\widehat{\rho}} - \frac{1}{\bar{\rho}}\right)\overline{\mathbf{m}}\overline{\mathbf{m}} - \bar{\rho}\,\tau^{\mathrm{SFS}}\,. \tag{5.3}$$

The term

$$\tau^{\mathrm{SFS}} \coloneqq \widehat{\mathbf{u}}\widetilde{\mathbf{u}} - \widetilde{\mathbf{u}}\widetilde{\mathbf{u}} \tag{5.4}$$

is known as the sub filter scale stress tensor. As the name suggests, it acts in a similar manner to the viscous stress, and it is usually modeled in terms of the resolved rate of strain. Similarly, for the advection term in the enthalpy equation,

$$\widetilde{h}\overline{\mathbf{m}} - \overline{h\mathbf{m}} = \bar{\rho}\left(\widetilde{h}\widetilde{\mathbf{u}} - \widetilde{h\mathbf{u}}\right) =: \mathbf{q}^{\mathrm{SFS}}\,, \tag{5.5}$$

which is called the sub filter scale heat flux[1], and it is usually modeled in terms of the resolved Fourier heat flux. The sub-filter stress (τ^{SFS}) and heat flux ($\mathbf{q}^{\mathrm{SFS}}$) play a central role in large eddy simulations of both incompressible and compressible flows.

Under the assumption that filtering commutes with spatial and temporal derivatives, the filtered transport equations are

$$\frac{\partial\widehat{\rho}}{\partial t} + \nabla\cdot\overline{\mathbf{m}} = R^{(0)}\,, \tag{5.6a}$$

$$\frac{\partial\overline{\mathbf{m}}}{\partial t} + \nabla\cdot\left(\widehat{\mathbf{u}}\;\overline{\mathbf{m}}\right) - \nabla\cdot\left(2\widehat{\mu}\widehat{S}\right) + \nabla\bar{p} - \bar{\mathbf{F}} = -\nabla\cdot\left(\bar{\rho}\tau^{\mathrm{SFS}}\right) + \nabla\cdot R^{(1)} + 2\nabla\cdot\left(R^{\mathrm{visc}} + A^{\mathrm{visc}} + B^{\mathrm{visc}}\right)\,, \tag{5.6b}$$

$$\frac{\partial\widehat{\rho}\widetilde{h}}{\partial t} + \nabla\cdot\left(\widetilde{h}\;\overline{\mathbf{m}}\right) - \nabla\cdot\left(\widehat{\rho\alpha}\;\nabla\widetilde{h}\right) - \bar{Q} = \nabla\cdot\mathbf{q}^{\mathrm{SFS}} + R^{(2)} + \nabla\cdot\left(\mathbf{A}^{\mathrm{fhf}} + \mathbf{B}^{\mathrm{fhf}}\right)\,. \tag{5.6c}$$

All non-computable (sub-filter) terms are gathered on the right-hand sides:

$$R^{(0)} \coloneqq \frac{\partial}{\partial t}\left(\widehat{\rho} - \bar{\rho}\right)\,, \tag{5.7a}$$

$$R^{(1)} \coloneqq \left(\frac{1}{\widehat{\rho}} - \frac{1}{\bar{\rho}}\right)\,\overline{\mathbf{m}}\overline{\mathbf{m}}\,, \tag{5.7b}$$

$$R^{(2)} \coloneqq \frac{\partial}{\partial t}\left(\left(\widehat{\rho} - \bar{\rho}\right)\widetilde{h}\right)\,, \tag{5.7c}$$

$$R^{\mathrm{visc}} \coloneqq \widehat{\mu}\left(\widetilde{S} - \widehat{S}\right)\,, \tag{5.7d}$$

$$A^{\mathrm{visc}} \coloneqq \bar{\rho}\left(\widetilde{\nu S} - \tilde{\nu}\widetilde{S}\right)\,, \tag{5.7e}$$

$$B^{\mathrm{visc}} \coloneqq \left(\bar{\mu} - \widehat{\mu}\right)\widetilde{S}\,, \tag{5.7f}$$

$$\mathbf{A}^{\mathrm{fhf}} \coloneqq \bar{\rho}\left(\widetilde{\alpha\nabla h} - \tilde{\alpha}\,\nabla\widetilde{h}\right)\,, \tag{5.7g}$$

$$\mathbf{B}^{\mathrm{fhf}} \coloneqq \left(\widehat{\rho\alpha} - \widehat{\rho}\widehat{\alpha}\right)\nabla\widetilde{h}\,. \tag{5.7h}$$

[1] Most other literature refers to $\bar{\rho}\left(\widetilde{T}\widetilde{\mathbf{u}} - \widetilde{T}\widetilde{\mathbf{u}}\right)$ as the sub-filter heat flux, or the sub-filter temperature flux. That definition works well when working with the ideal gas law.

The sub-filter terms due to the viscous stress sum to

$$\begin{aligned} R^{\mathrm{visc}} + A^{\mathrm{visc}} + B^{\mathrm{visc}} &= \widehat{\mu}\left(\widetilde{S} - \widehat{S}\right) \quad + \quad \overline{\rho v S} - \overline{\rho v}\,\widetilde{S} \quad + \quad \left(\overline{\mu} - \widehat{\mu}\right)\widetilde{S} \\ &= \overline{\mu S} - \widehat{\mu}\widehat{S} \end{aligned} \tag{5.8}$$

to account for the difference between the filtered original $\overline{\nabla \cdot (\mu S)}$ and the computable $\nabla \cdot \left(\widehat{\mu}\widehat{S}\right)$. Similarly, the sub-filter terms due to the Fourier heat flux ('fhf') sum to

$$\mathbf{A}^{\mathrm{fhf}} + \mathbf{B}^{\mathrm{fhf}} = \overline{\rho\alpha\nabla h} - \widehat{\rho\alpha}\nabla\widetilde{h} \ . \tag{5.9}$$

The terms in Eqs. 5.4, 5.5, and 5.7a–5.7h differ strongly in magnitude, so some can be neglected.

Most other authors have considered the right-hand side of Eq. 5.8 (resp. Eq. 5.9) as a single sub-filter term, rather than splitting it into several terms as is done here. The present decomposition shows more explicitly that each term in Eqs. 5.7a–5.7h falls into one of the following categories.

1. A^{visc} and $\mathbf{A}^{\mathrm{fhf}}$ are commutation errors between the Favre filter, and multiplication by a transport property. These occur in LES with inhomogeneous transport properties. It is standard practice to neglect them (see, e.g., the discussions in [1, 2]).

2. B^{visc} and $\mathbf{B}^{\mathrm{fhf}}$ are due to the difference between the filtered and the computable transport properties. It is standard practice to neglect them (e.g., [3]).

3. The terms R^0, R^1, R^2, and R^{visc} result from the difference between the computable density and the filtered density. For R^{visc} this can be made clear by rewriting it as

$$R^{\mathrm{visc}} = \widehat{\mu}\ L\left[\nabla\left(\widetilde{\mathbf{u}} - \widehat{\mathbf{u}}\right)\right] = \widehat{\mu}\ L\left[\nabla\left(\left(\frac{1}{\overline{\rho}} - \frac{1}{\widehat{\rho}}\right)\overline{\mathbf{m}}\right)\right] \ . \tag{5.10}$$

   These terms are particular to LES of low-Mach flows; they do not occur in incompressible flows (where ρ is usually constant), or in high-Mach compressible flows (where $\bar{\rho}$ is computable as one the resolved quantities).

   We strongly suspect that these terms can be neglected, in analogy to those in category 2.

4. The turbulent stress tensor τ^{SFS} and the sub-filter temperature flux $\mathbf{q}^{\mathrm{SFS}}$ are both commutation errors between the Favre filter, and multiplication by an advecting field.

In other words, all terms are neglected in low-Mach flows, except those in the last category, which also dominate in incompressible flows with constant fluid properties.

5.2. Sub-filter Scale models

The oldest and most common form of LES is to model the effect of the sub-filter stress tensor τ^{SFS} (Eq. 5.4) as a function of the resolved quantities, which Sagaut [4] calls a 'functional' model. In particular, this is usually done in terms of the resolved rate of strain, so that

$$-\nabla\cdot\left(\bar{\rho}\tau^{\mathrm{SFS}}\right)\approx\nabla\cdot\left(2\mu^{\mathrm{sfs}}\hat{S}\right)\ . \tag{5.11}$$

This has the same form as the viscous stress, and is therefore easy to implement: the momentum equation is the same as for direct numerical simulation, except that the kinematic viscosity becomes

$$\nu=\nu^{\mathrm{molec}}+\nu^{\mathrm{sfs}}\ , \tag{5.12}$$

where ν^{molec} is the 'molecular' (i.e., actual, physical) viscosity, and ν^{sfs} is the extra viscous effect ('eddy viscosity') due to flow structures that are smaller than the LES filter width.

The main challenge is to model ν^{sfs} in terms of computable quantities. Obviously it cannot be constant, because that would be come down to simulating the equivalent flow at a lower Reynolds number. Instead ν^{sfs} is expressed in terms of the velocity gradient. Since the fluid is assumed to have no memory, ν^{sfs} only depends on the local value of $\nabla\mathbf{u}$. As there is no clear best choice, we mention several models below.

The sub-filter viscosity depends on the symmetric part of the computable velocity gradient, given by

$$\mathcal{S}_{ij}\coloneqq\frac{1}{2}\left(\nabla_i\widehat{u}_j+\nabla_j\widehat{u}_i\right)\ . \tag{5.13}$$

This is not trace-free, but otherwise it is equivalent to the resolved rate of strain $\hat{S}$ (Eq. 5.1). The anti-symmetric part of the velocity gradient corresponds to rigid rotation, in which the relative position of the fluid particles does not change, and which is assumed not to result in turbulent dissipation, and therefore not to affect ν^{sfs}.

Since the sub-filter stress is supposed to model the physical phenomenon of turbulent dissipation, it stands to reason that ν^{sfs} be a function of the invariants of $\mathcal{S}$. The main invariants of an arbitrary rank-2 tensor A in three dimensions with eigenvalues λ_i are $\mathrm{trace}\,(A)=\lambda_1+\lambda_2+\lambda_3$, $\mathrm{trace}\left(A^2\right)=\lambda_1^2+\lambda_2^2+\lambda_3^2$, and $\mathrm{trace}\left(A^3\right)=\lambda_1^3+\lambda_2^3+\lambda_3^3$. The numerical velocity in an incompressible flow is either exactly solenoidal, or very close to solenoidal (depending on the spatial discretization). Even in low-Mach number compressible flows, we can assume that $\mathrm{trace}\,(\mathcal{S})=\nabla\cdot\mathbf{u}$ has a much smaller impact on the turbulent dissipation than the deviatoric part of $\mathcal{S}$, and in any case most LES models for compressible flows are straightforward extensions of incompressible models, so the first invariant does not play a role.

The oldest and simplest model therefore expresses ν^{sfs} in terms of the second main invariant of $\mathcal{S}$:

$$\nu^{\mathrm{sfs}}_{\mathrm{smag}}=A^2_{\mathrm{smag}}\sqrt{2\ \mathrm{trace}\,(\mathcal{S}^2)}=A^2_{\mathrm{smag}}\sqrt{2\mathcal{S}_{kl}\mathcal{S}_{kl}}=A^2_{\mathrm{smag}}\sqrt{2}\left\|\mathcal{S}\right\|_{\mathrm{Frob}}\ . \tag{5.14}$$

Here $\|\cdot\|_{\text{Frob}}$ is the Frobenius norm. This is known as the Smagorinsky model, and it the most well known, the most researched, and probably still the most commonly used.

The factor A_{smag} still needs to be determined. It has the dimension of length, and it is directly related to the filter width: larger values of A_{smag} damp more of the small scales of the flow, lowering the maximum frequency in the Fourier transform of the computed velocity. This is made explicit by writing

$$A_{\text{smag}} = C_{\text{smag}}\,\Delta\,, \tag{5.15}$$

where Δ is the LES filter width, and C_{smag} is known as the Smagorinsky constant. In practice Δ is usually coupled to the resolution of the spatial discretization. Given a characteristic element length ℓ, the universally accepted engineering practice is to let $\Delta = 2\ell$. This typically results in both a modelling error (due to the sub-filter flow structures, which are smaller than Δ), and a discretization error (because ℓ is not much smaller than Δ).

A theoretical analysis of isotropic turbulence suggests that the Smagorinsky constant is indeed the same for all flows [5]. Berselli *et al.* [6, pp. 71–77] estimate it at $C_{\text{smag}} \approx 0.17$, whereas Sagaut [4, pp. 113–124] gives $C_{\text{smag}} \approx 0.148$ or $C_{\text{smag}} \approx 0.18$. Extensive numerical experience has shown that these values are too large for almost all flows and discretizations; it is too dissipative. Typically C_{smag} is lowered to approximately 0.1 to achieve the right amount of total kinetic energy removal, though this does not guarantee an optimal local structure of turbulent dissipation.

Another shortcoming of the Smagorinsky model is that the dissipation does not automatically vanish in the laminar layer near the wall. One possible solution is to multiply A_{smag} by a scaling factor. By far the most common is the Van Driest damping function

$$f\,(y^+) = 1 - e^{-y^+/25}\,, \tag{5.16}$$

where y^+ is the dimensionless wall distance, based on the wall shear stress (as defined in Eq. 5.32 below). Another way of looking at this is that the LES method implies a spatial filter through convolution with a filtering kernel, but a symmetric kernel is not possible at the wall. The filter width is therefore gradually reduced to zero near the wall.

It can be shown theoretically that $\nu^{\text{sfs}} \propto (y^+)^3$, though the above Van Driest damping produces $\nu^{\text{sfs}} \propto (y^+)^2$. Piomelli *et al.* [7] suggested a different damping function that achieves the correct asymptotic behavior near the wall, but found no significant improvement over Van Driest damping. See Inagaki [8] for a more recent review.

Nicoud and Ducros [9] have argued that the Smagorinsky model is fundamentally flawed because it does not include the effect of rotation, and proposed the alternative WALE (wall-adaptive local eddy) viscosity model, which is based on the square of the velocity gradient, that is,

$$\left(B^2\right)_{ij} \coloneqq \left(\nabla_i u_k\right)\left(\nabla_k u_j\right)\,. \tag{5.17}$$

Denoting

$$F \coloneqq L\left[B^2\right] , \tag{5.18}$$

we have

$$\nu_{\mathrm{WALE}}^{\mathrm{sfs}} = A_{\mathrm{WALE}}^2 \frac{\left(F_{kl}F_{kl}\right)^{3/2}}{\left(\mathcal{S}_{kl}\mathcal{S}_{kl}\right)^{5/2} + \left(F_{kl}F_{kl}\right)^{5/4}} = A_{\mathrm{WALE}}^2 \frac{\|F\|_{\mathrm{Frob}}^3}{\|\mathcal{S}\|_{\mathrm{Frob}}^5 + \|F\|_{\mathrm{Frob}}^{5/2}} . \tag{5.19}$$

The idea is that $\|F\|_{\mathrm{Frob}}$ and $\|S\|_{\mathrm{Frob}}$ scale differently near the wall, so it is possible to choose the exponents in Eq. 5.19 in such a way that we get the appropriate $\nu_{\mathrm{WALE}}^{\mathrm{sfs}} = \mathcal{O}((y^+)^3)$ behavior. The term $\|F\|_{\mathrm{Frob}}^{5/2}$ is only there to ensure that the denominator is nonzero, giving the expression a somewhat artificial look, though it has performed well in practice. (See [10] for an example with a discontinuous Galerkin method.)

In analogy to Eq. 5.15, the prefactor is related to the filter width:

$$A_{\mathrm{WALE}} = C_{\mathrm{WALE}}\, \Delta . \tag{5.20}$$

Nicoud and Ducros [9] chose the constant C_{WALE} such that Eqs. 5.14 and 5.19 predict the same amount of turbulent dissipation in homogeneous, isotropic turbulence. This calibration gave $C_{\mathrm{WALE}} \approx 0.5$. Garnier *et al.* [3, p. 88] later found $\left(C_{\mathrm{WALE}}/C_{\mathrm{smag}}\right)^2 \approx 10.6$, which comes down to $C_{\mathrm{WALE}} \in (0.326, 0.59)$ for $C_{\mathrm{smag}} \in (0.1, 0.18)$. The widely used ANSYS-Fluent [11] software package sets $C_{\mathrm{WALE}} = 0.325$ by default.

Finally, we consider the QR model, which has been introduced far more recently than the Smagorinsky and the WALE models by Verstappen [12] in 2011. Rather than focussing on physical reasoning, he assumed that the sub-filter scale flow should not influence the larger structures, and then looked for the minimal value of ν^{sfs} that meets this constraint. This involved an interesting analysis of the filtered Navier-Stokes equations, resulting in[2]

$$\nu_{\mathrm{QR}}^{\mathrm{sfs}} = A_{\mathrm{QR}}^2 \frac{\left|\mathcal{S}_{kl}\mathcal{S}_{lm}\mathcal{S}_{mk}\right|}{\mathcal{S}_{kl}\mathcal{S}_{kl}} = A_{\mathrm{QR}}^2 \left|\frac{\mathrm{trace}\left(\mathcal{S}^3\right)}{\mathrm{trace}\left(\mathcal{S}^2\right)}\right| , \tag{5.21}$$

making it the only model in this section that involves the third invariant of $\mathcal{S}$. As before, we can write

$$A_{\mathrm{QR}} = C_{\mathrm{QR}}\, \Delta . \tag{5.22}$$

Verstappen [12] finds $C_{\mathrm{QR}} = 1/\pi = 0.32$. The sub-filter viscosity has proper $\mathcal{O}((y^+)^3)$ behavior near the wall for constant C_{QR}.

The prefactors (A_{smag}, A_{WALE}, A_{QR}) in Eqs. 5.14, 5.19, 5.21 can also be determined from the simulation itself, rather than depending on user-defined parameters. This is done with the so-called 'dynamic model', originally introduced for the Smagorinsky model by Germano *et al.* [13], who projected the numerical solution

[2] Note that the trace is the sum of the eigenvalues and $\mathcal{S}$ is symmetric, so $\mathrm{trace}\left(\mathcal{S}^2\right) \geq 0$. The denominator could theoretically vanish, but $\nu_{\mathrm{QR}}^{\mathrm{sfs}}$ is well behaved in the limit $S_{ij} \to 0$.

onto a coarser solution space (e.g., a coarser grid). The LES model (in their case Eq. 5.14) is assumed to be valid on both the fine and coarse grids. Since the prefactor A is assumed not to change between these two flow scales, it can be computed by comparing the coarse and fine solutions. The dynamic Smagorinsky model displays the correct $\mathcal{O}\left(\left(y^+\right)^3\right)$ asymptotic behavior, and has widely been found to yield far better results than the 'constant' Smagorinsky model (i.e., with a user-defined constant). (See, e.g., the comparisons and the very clear discussion in Vreman [14], who also studied compressible flows.)

In the context of a high-order finite element discretization, the projection onto a coarse grid could be replaced by a projection onto a lower-order polynomial space. This has been done with a discontinuous Galerkin method by Abbà *et al.* [15], though it is not clear whether this is better than projecting onto larger elements. It has also been implemented in `DGFlows`, but we do not use it in this chapter. These different types of projections are reminiscent of p-multigrid and h-multigrid solvers, which were mentioned in section 2.6.

A dynamic model also makes it easier to use a symmetric positive definite matrix instead of a scalar for μ^{sfs} in Eq. 5.11, which could in principal handle anisotropic turbulence better. Abbà *et al.* [15] found that an anisotropic μ^{sfs} tensor indeed gives superior results for in wall-bounded flow. Nevertheless using a scalar μ^{sfs} is far more common.

For the sub-filter heat flux we only consider the simplest model, which is based on a turbulent Prandtl number Pr_t that relates the effect of $\mathbf{q}^{\text{SFS}}$ to $\boldsymbol{\tau}^{\text{SFS}}$. First we assume that the sub-filter heat flux results in a net turbulent dissipation, analogous to Eq. 5.11:

$$\nabla \cdot \mathbf{q}^{\text{SFS}} \approx \nabla \cdot \left((\rho\alpha)^{\text{sfs}} \nabla \widehat{h} \right) . \tag{5.23}$$

Then we express $(\rho\alpha)^{\text{sfs}}$ in terms of ν^{sfs} with a turbulent Prandtl number:

$$(\rho\alpha)^{\text{sfs}} = \frac{\widehat{\rho}\, \nu^{\text{sfs}}}{\text{Pr}_\text{t}} , \tag{5.24}$$

where we have used the computable density $\widehat{\rho}$ because that is the only directly available value in a low-Mach number simulation. This is easy to implement: compared to direct numerical simulation, only the fluid property $k/c_p = \rho\alpha$ needs to be adjusted to

$$\rho\alpha = (\rho\alpha)^{\text{molec}} + (\rho\alpha)^{\text{sfs}} , \tag{5.25}$$

where $(\rho\alpha)^{\text{molec}}$ is the 'molecular' (physical) fluid property.

The turbulent Prandtl number is usually chosen in the range $(0.3, 0.9)$ [3, p. 84]. Lesieur [16] derived a theoretical value of $\text{Pr}_\text{t} = 0.6$. It can also be estimated with a dynamic procedure.

5.3. Numerical Simulation

5.3.1. Discretization

The spatial discretization is as described in the previous chapters, except for the pressure stabilization for equal-order polynomial spaces for the pressure and the

mass flux. The problem with the penalty term in the pressure equation (a^{stab} in Eq. 2.7) is that it makes the linear system too stiff, and our implementation `DGFLows` is not well-equipped to deal with this. See section 3.5.1 for a discussion of this problem.

The calculations with equal-order discretizations in this chapter are therefore stabilized by penalizing violations of the continuity equation, as has been suggested by Krank *et al.* [17]. (See also the remarks in section 2.2.) We incorporate their penalty terms into the momentum equation, so that we do not require a separate projection step for the momentum, as they do. That is, the terms

$$a^*(\mathbf{w},\mathbf{v}) = \sum_{T\in\mathcal{T}} \int_T \theta\, \ell^T\, (\nabla\cdot\mathbf{w})\,(\nabla\cdot\mathbf{v}) + \sum_{F\in\mathcal{F}^{\mathrm{D,i}}} \int_F \{\theta\}\,[\![\mathbf{n}^F\cdot\mathbf{w}]\!]\,[\![\mathbf{n}^F\cdot\mathbf{v}]\!] \tag{5.26}$$

and

$$l^*(\mathbf{v}) = \sum_{F\in\mathcal{F}^{\mathrm{D}}} \int_F \theta\,(\mathbf{n}\cdot\mathbf{v})\left(\mathbf{n}\cdot\mathbf{m}^{\mathrm{D}}\right) \tag{5.27}$$

are added to the bilinear and linear operators in the discrete momentum equation (given by Eq. 2.10 for time-independent flows). Here ℓ^T is a characteristic length of the element; we use the value

$$\ell^T = \frac{\|T\|_{\text{leb}}^{1/d}}{1+\mathcal{P}_m} \tag{5.28}$$

for a d-dimensional element with polynomial order $\mathcal{P}_m$ for the mass flux, and do not investigate other estimates.

Note that a^* and l^* couple the directions of the mass flux. Previously they were only coupled implicitly in the pressure projection (and, less importantly, in the viscous stress). We solve the momentum equation for all directions simultaneously.

The penalty parameter θ has the same dimension as the velocity, and Krank *et al.* [17] used the natural $\theta = |\mathbf{u}| \coloneqq \sqrt{\mathbf{u}\cdot\mathbf{u}}$. This makes a^* and l^* nonlinear in $\mathbf{u}$, which could presumably be handled efficiently by basing θ on an extrapolation of $\mathbf{u}$ from previous time steps. More problematically, the term a^* would need to be reassembled at every time step. In our implementation this takes a significant part of the total calculation time.

We therefore use the alternative form

$$\theta = \langle|\mathbf{u}|\rangle\ , \tag{5.29}$$

where $\langle\cdot\rangle$ denotes the average over time and over the homogenous directions. This means that θ is time-independent, so that a^* and l^* can be precomputed. In Eq. 5.26 the penalty parameter is an average of the neighbors of the faces, but taking the maximum value would also have seemed reasonable to us. For the present calculations it makes no difference, since the above form of θ is virtually continuous at the element boundaries.

5.3.2. Including a Variable Density

This chapter will only present a simulation with a constant density. As explained in section 5.1, LES of low-Mach number flow requires models for the same sub-filter scale terms as for constant-density flows. The results for incompressible flows are therefore directly relevant for variable-density flows.

It would nonetheless have been interesting to include a variable-density test case. A low-Mach number flow would pose new challenges, such as a stronger coupling between the sub-filter heat flux and the momentum equation, due to the temperature-dependent density. Unfortunately our limited computational resources do not allow for this.

The problem is that the transport equations in the form of Eqs. 1.1a–1.1c with a non-constant ρ are only well-posed when there is an outflow boundary condition. This would require an inlet with an expensive turbulent inflow generator. The simulation must furthermore describe a developing flow, which requires a far larger domain than when there are periodic boundary conditions. Nemati [18] has nevertheless simulated developing low-Mach number flow. He used a specialized implementation that depends on the specific geometry of pipe flows, which have one main flow direction, and where solving the Poisson equation might have been sped up with a fast Fourier transform.

Of course a variable-density flow can also be simulated with a compressible solver at a low, nonzero Mach number, though this is probably far more expensive than using the transport equations in the low-Mach number limit. Examples include [19], [20], and [21], who all required generic massively parallel solvers with some of the world's largest supercomputers. These authors performed direct numerical simulation of CO_2 at a supercritical pressure.

A computationally more efficient approach is to introduce a variable thermodynamic pressure p^{th}. The density then becomes a function of both the temperature T and p^{th}, which makes the governing equations well posed with periodic boundary conditions (i.e., a closed domain). The enthalpy equation in low-Mach number limit gets an extra term $\mathrm{d}p^{\mathrm{th}}/\mathrm{d}t$, as explained in the seminal paper by Rehm and Baum [22]. The thermodynamic pressure is updated after each time step. Nicoud [23] was one of the first to present a direct numerical simulation of variable-density flow in this manner. See also the review in Knikker [24]. Avila *et al.* [25] have also included an LES model with the variable density.

These authors have all assumed an ideal gas. This is almost always done in the literature, though a notable exception is the work by Accary *et al.* [26, 27], who used a more general Van der Waals equation. We are not aware of similar papers with an arbitrary equation of state.

A dubious alternative approach to cheap variable-density flow calculations was taken by surprisingly many authors [28–30], who simulated supercritical fluids in closed domains, but ignored the effect of the thermodynamic pressure on the density. That is, they left out the $\mathrm{d}p^{\mathrm{th}}/\mathrm{d}t$ term in the enthalpy equation, and computed the equation of state at a fixed p^{th}, so that ρ was merely a function of h. This results in a mathematically ill-posed problem, because the continuity equation implies a constant the total fluid mass (due to the closed domain), whereas the enthalpy

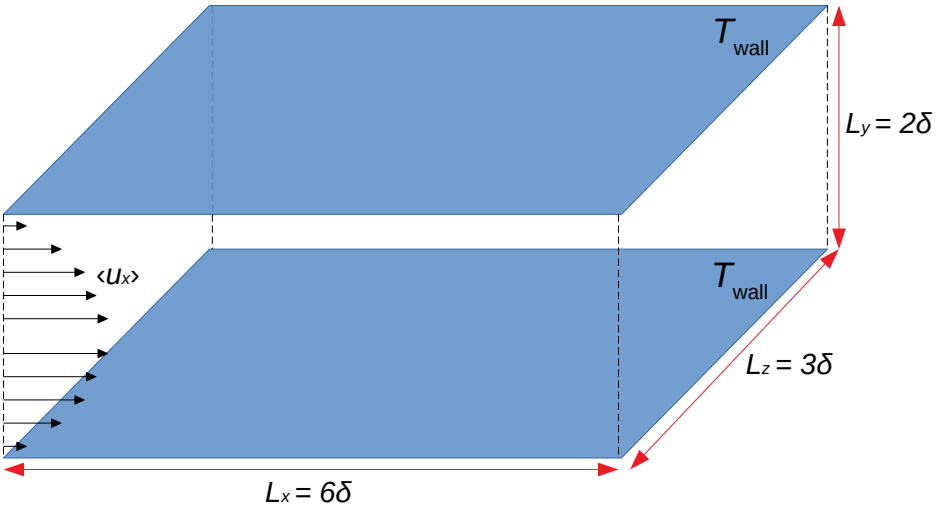


Figure 5.1: Geometry of the infinite plane channel.

equation implies a fluctuating total mass (due to the imposed heat flux and the fact that $\rho = \rho(h)$). The results in these papers are therefore fundamentally irreproducible.

The authors have in common that they focused on physical interpretations of the CFD results, rather than on the numerical method. Their computations were probably helped by the fact that they used finite difference schemes, which do not conserve the global mass. In one case [30], we know from a private conversation with one of the authors that the simulations were kick-started with an artificial mass sink in the continuity equation, to allow for the average temperature to rise after the initial condition. It is possible that the conclusions in [30] hold despite the mathematical inconsistency, because they are based on local fluctuations, which are perhaps not strongly influenced by the global conservation of mass.

5.4. Test Case: Infinite Plane Channel Flow

This section describes a numerical simulation of flow between two planes. The geometry is shown in Fig. 5.1. The origin is placed between the two planes, so that $-\delta < y < \delta$. The boundaries are periodic in the x- and z-directions to approximate flow between infinite planes.

The flow is driven by a homogeneous, constant volumetric force $\mathbf{F} = [F_x, 0, 0]$ and volumetric heat source Q. The top and bottom planes have Dirichlet boundary conditions for the velocity ($\mathbf{u}|_{y=\pm\delta} = \mathbf{0}$) and the temperature ($T|_{y=\pm\delta} = T_{\text{wall}}$). The setup and the time-averaged quantities are therefore symmetrical about $y = 0$, and homogeneous in the x- and z-directions.

5.4.1. Dimensionless Analysis

The following dimensionless quantities are defined in the usual way, based on the average shear stress τ_{wall} at the wall. The characteristic friction velocity at the wall is $u_\tau := \sqrt{\tau_{\text{wall}}/\rho}$, and from that follows

$$\text{the friction Reynolds number: } \mathrm{Re}_\tau := u_\tau R_\text{h}/\nu = u_\tau \delta/\nu \; ; \tag{5.30}$$

$$\text{the turn-over time: } t^+ := t u_\tau/\delta \; ; \tag{5.31}$$

$$\text{the wall distance: } y^+ := y^* u_\tau/\nu = (y^*/\delta)\mathrm{Re}_\tau \; , \tag{5.32}$$

where

$$R_\text{h} := \frac{\text{flow area}}{\text{wetted perimeter}} = \delta \tag{5.33}$$

is the hydraulic radius, and $y^* = \delta - \mathrm{abs}(y)$ is the dimensional distance from the wall.

The desired Re_τ is chosen as a free parameter. Since the average wall friction τ_{wall} balances the volumetric force, this fixes the volumetric force at $F_x = \tau_{\text{wall}}/R_\text{h} = \tau_{\text{wall}}/\delta$. The average velocity in the domain is therefore a result of the computation, not an input parameter.

Many authors have taken the opposite approach by choosing a bulk Reynolds number

$$\mathrm{Re}_{\text{bulk}} := \frac{D_\text{h} u_{\text{bulk}}}{\nu} = \frac{4 R_\text{h} u_{\text{bulk}}}{\nu} \; , \tag{5.34}$$

which is based on the hydraulic diameter $D_\text{h} := 4R_\text{h}$ and the time-averaged bulk velocity

$$u_{\text{bulk}} := \frac{1}{\|\Omega\|_{\text{leb}}} \int_\Omega u_1 = \frac{1}{2\delta} \int_{|y|<\delta} u_1 \; . \tag{5.35}$$

The volumetric force is then adjusted dynamically during the simulation to get the desired $\mathrm{Re}_{\text{bulk}}$. The disadvantage of that approach is that the results are less reproducible, because the actual imposed force F_x is unknown.

The two Reynolds numbers Re_τ and $\mathrm{Re}_{\text{bulk}}$ are related by the Darcy-Weisbach equation

$$F_x = f_\text{D} \frac{\rho}{2 D_\text{h}} u_{\text{bulk}}^2 \; , \tag{5.36}$$

where f_D is the Darcy friction factor. Dimensionless analysis shows that f_D is only a function of $\mathrm{Re}_{\text{bulk}}$ (e.g., [31, pp. 177–184]). The friction Reynolds number can be expressed in terms of the volumetric driving force as

$$\mathrm{Re}_\tau := \frac{R_\text{h}\rho}{\mu} u_\tau = \frac{R_\text{h}\rho}{\mu} \sqrt{\frac{\tau_{\text{wall}}}{\rho}} = \frac{R_\text{h}\rho}{\mu} \sqrt{\frac{R_\text{h} F_x}{\rho}} \; . \tag{5.37}$$

Substituting the Darcy-Weisbach equation and the definition for the bulk Reynolds number gives $\mathrm{Re}_\tau = \mathrm{Re}_{\text{bulk}} \, (R_\text{h}/D_\text{h}) \sqrt{(R_\text{h}/D_\text{h}) \, (f_\text{D}/2)}$, or

$$\mathrm{Re}_{\text{bulk}} = \sqrt{\frac{128}{f_\text{D}}} \, \mathrm{Re}_\tau \approx \frac{11.3}{\sqrt{f_\text{D}}} \, \mathrm{Re}_\tau \; . \tag{5.38}$$

We have surprisingly not seen this relation in previous literature, although it holds for all channel geometries.

The friction factor can be estimated by using measurement data from turbulent pipe flow. If the Reynolds number is sufficiently large, then the curvature of the wall is negligible compared to the size of the flow structures near the wall, which justifies a comparison between channels of different geometries. The semi-empirical Colebrook-White correlation for smooth circular pipes is[3]

$$\frac{1}{\sqrt{f_\mathrm{D}}} = -2\log_{10}\left(\frac{A}{\sqrt{f_\mathrm{D}}\,\mathrm{Re}_\mathrm{bulk}}\right) = -2\log_{10}\left(\frac{A/\sqrt{128}}{\mathrm{Re}_\tau}\right) \tag{5.39}$$

with $A = 2.51$. This is valid for $\mathrm{Re}_\mathrm{bulk} > 4.0 \cdot 10^3$, or, equivalently, $\mathrm{Re}_\tau > 71$.

5.4.2. Initial Condition

We are interested in turbulent flow, which is not obtained with all initial conditions. A turbulent channel flow can only be sustained above a certain critical Reynolds number, whereas the laminar solution

$$\begin{cases} \mathbf{u}^\mathrm{laminar} = u_\mathrm{bulk}\left[\frac{3}{2}\left(1-\left(\frac{y}{\delta}\right)^2\right),\ 0,\ 0\right] \\ T^\mathrm{laminar} = T_\mathrm{wall} + \frac{\delta^2 Q}{2k}\left(1-\left(\frac{y}{\delta}\right)^2\right) \end{cases} \tag{5.40}$$

is stable at all Reynolds numbers, making it unsuitable as an initial condition.

A standard approach is to perturb the laminar solution vector with a random number generator. For the present discretization, such a perturbation is quickly damped by the viscous stress, because many degrees of freedom correspond to high-order basis functions. This phenomenon of laminarization after a high-frequency perturbation has been well known in the CFD community for a long time (e.g., [33]), and has more recently also been established experimentally [34].

The laminar velocity field is therefore perturbed by both a random solution vector, and the arbitrarily chosen large-scale solenoidal structure

$$0.10\ u_\mathrm{bulk}\frac{3}{2}\left(1-\left(\frac{y}{\delta}\right)^2\right)\begin{bmatrix}\sin(2\pi z)\sin(y/\delta)\\ 0\\ \sin(2\pi x)\end{bmatrix}, \tag{5.41}$$

which induces a turbulent solution.

5.4.3. Domain Size and Mesh

The domain should be large enough to encompass the largest physical structures of the flow between two infinite planes. This can be checked a posteriori by investigating the correlation between the turbulent fluctuations at various points in the

[3] There are many other commonly used correlations, but they have no independent value, as they are designed to approximate the Colebrook-White correlation. See, e.g., the discussion in Zigrang and Sylvester [32].

domain. For any two points that are half a domain size apart in the x- or z-direction, the correlation should be negligible.

Our domain size is based in part on the correlation coefficients that are reported in [35] and [36]. Piomelli *et al.* [37] investigated various domain sizes more systematically, and concluded that $(L_x, L_y, L_z) = (6\delta, 2\delta, 3\delta)$ is 'marginally sufficient' for second-order turbulent statistics at our Reynolds number, so this is what we use.

Our setup is meant to reproduce one of the test cases in Patel *et al.* [30], who performed direct numerical simulation (DNS) at $\mathrm{Re}_\tau = 395$. This is a popular Reynolds number, presumably because it was used in the landmark paper by Moser *et al.* [36]. Following Patel *et al.* , the fluid properties are kept constant, and $\mathrm{Pr} = 1$.

The element sizes in the inhomogeneous y-direction are based on a $\tanh$ grid spacing. More precisely, the element boundaries are located at $y/\delta \in \{\xi_i\}_{i=0}^{N}$ with

$$\xi_i = \frac{\tanh\left(\gamma\left(\frac{2i}{N} - 1\right)\right)}{\tanh\gamma} \,. \tag{5.42}$$

Here $\gamma > 0$ is a stretching parameter, with $\gamma \to 0$ corresponding to a uniform element size. The parameters γ and N are fixed implicitly by choosing the minimum and maximum element widths ($\Delta y^{\min}$ and $\Delta y^{\max}$) at the wall and in the center of the channel.[4]

We place the first element boundary at $y^+ = 2$. Other authors have suggested putting the first grid point at approximately $y^+ = 1$ or $y^+ = 2$ when using a high-order finite-difference scheme, which normally cannot attain its full order of accuracy near the wall. It is unclear whether a discontinuous Galerkin method permits a larger wall element width, though de Wiart *et al.* [38] have obtained accurate results with wall elements at $y^+ = 2.5$, using a third-order tensor-product polynomial space. We set $N = 46$ and $\gamma = 2.2132$, so that the first element boundary is at $y^+ = 2.00$, and the maximum element width is 38.8 wall units.

The element widths in the homogeneous x- and z-directions are constant throughout the domain. This is by far the most common in the literature, though Collis [39] has argued that the geometric flexibility of the discontinuous Galerkin method should be used to refine the elements at the wall in the x- and z-directions, so that their aspect ratios do not become too large. Previous authors have used many

[4] This entails solving the nonlinear coupled equations $\xi_1 - \xi_0 = \Delta y^{\min}$ and $\xi_m - \xi_{m-1} = \Delta y^{\max}$ with $m \coloneqq \lceil N/2 \rceil$ for (γ, N). By assuming $N \gg 1$ (which is normally the case), the index i can be treated as a continuous variable, giving the estimates

$$\Delta y^{\min} \approx \left.\frac{\partial \xi_i}{\partial i}\right|_{i=0} = \frac{2\gamma/N}{\tanh\gamma}\,\frac{1}{\cosh^2\gamma} \quad \text{and} \quad \Delta y^{\max} \approx \left.\frac{\partial \xi_i}{\partial i}\right|_{i=N/2} = \frac{2\gamma/N}{\tanh\gamma} \,, \tag{5.43}$$

resulting in a ratio of the element widths of

$$\frac{y^{\max}}{y^{\min}} = \cosh^2\gamma + \mathcal{O}\left(\frac{1}{N}\right) . \tag{5.44}$$

We found that $\gamma \approx \cosh^{-1}\sqrt{y^{\max}/y^{\min}}$ is very close to the actual solution for all $(y^{\min}, y^{\max})$, and that a fixed-point iteration always converges with this initial guess.

different aspect ratios for the minimum elements in the bulk; we use $\Delta z/\Delta y = 1$ and $\Delta x/\Delta z = 2$, resulting in element widths in the x- and z-directions of 77.6 and 38.8, as measured in wall units.

5.4.4. Results

We are interested in the ensemble averages of various quantities over many realizations of the flow, which are called the Reynolds averages, denoted by angle brackets (e.g., $\langle T \rangle$, $\langle m_1 \rangle$, …). The turbulent fluctuation is defined as the instantaneous deviation from the Reynolds average, that is,

$$\phi' \coloneqq \phi - \langle \phi \rangle \ . \tag{5.45}$$

In practice the ensemble average is assumed equal to the time average, which can be estimated from a single computation or experiment that runs for a sufficiently long time.

Besides the first-order statistics $\langle u_i \rangle$ and $\langle T \rangle$, we are also interested in the Reynolds stress

$$\left\langle u_i' u_j' \right\rangle = \langle u_i u_j \rangle - \langle u_i \rangle \langle u_j \rangle \ , \tag{5.46}$$

and the turbulent heat flux

$$\left\langle u_i' T' \right\rangle = \langle u_i T \rangle - \langle u_i \rangle \langle T \rangle \ . \tag{5.47}$$

These second-order statistics are commutation errors between the Reynolds average and multiplication, so they come up when Reynolds-averaging the convective terms in the incompressible Navier-Stokes equation.

The temperature can be made dimensionless in various ways; we use the reference value

$$T^{\mathrm{ref}} \coloneqq \mathrm{Re}_\tau^{-1} \frac{\delta^2 Q}{k} \ . \tag{5.48}$$

The group $\delta^2 Q/k$ contains diffusion-related quantities, and can also be seen in the laminar solution (Eq. 5.40). The factor Re_τ^{-1} takes convective heat transfer into account, and is meant to let $\left\langle T/T^{\mathrm{ref}} \right\rangle$ depend less on the Reynolds number, since $\langle T \rangle$ decreases with increasing Re_τ.

For our test case the quantities can also be averaged over the homogeneous directions, making the averages converge far more quickly. The quantities of interest are sampled every 20 time steps. They are projected onto the eighth-order polynomial space in the y coordinate in each element, so that no information is lost. The averages are based on approximately 10 turn-over times.

We also average the two values in the lower and upper halves of the domain (i.e., where $y < 0$ or $y > 0$). This does not work for the stresses involving u_2 (i.e., $\left\langle u_1' u_2' \right\rangle$ and $\left\langle T' u_2' \right\rangle$), because they are antisymmetric about $y = 0$. Therefore we will instead report values based on

$$u_{2\uparrow} \coloneqq u_2 \ \mathrm{sign}(y) = \pm u_2 \quad \text{for } \pm y > 0, \tag{5.49}$$

which is the component of the velocity toward the closest wall. This yields symmetric stresses $\langle u_1' u_{2\uparrow}' \rangle$ and $\langle T' u_{2\uparrow}' \rangle$. These could also be interpreted as the usual stresses $\langle u_1' u_2' \rangle$ and $\langle T' u_2' \rangle$ in the upper half of the domain.

Figure 5.2 shows the error in the average velocity profiles for a model-free high-order simulation (uDNS_{32}), a coarser model-free lower-order simulation (uDNS_{22}), and two LES simulations based on the coarser discretization (WALE and Smag.). The time step size is given by $(u_\tau/\delta)\delta t = 10^{-3}$, resulting in a maximum CFL number of approximately 0.7 in the flow direction on the coarse grid. The error is based on the DNS data in Moser *et al.* [36]. The figure also shows that the difference with the DNS data in Patel *et al.* [30] is negligible.

For comparison, the results of the LES by Singh, You, and Bose [40] are also shown (indicated by SYB). They varied the spatial discretization independently from the LES filter width, resulting in a grid-independent (i.e., fully resolved) LES. They used a dynamic Smagorinsky model with LES filter widths in the x-, y-, and z-directions of (78, 0.53–60, 39) (measured in wall units). Ours are comparable: approximately (72, 1.9–36, 36). We find that their results are approximately as accurate as our coarse-grid calculations, regardless of whether we use an LES model, and despite the fact that we do have a discretization error.

Figure 5.3 shows the averaged sub-filter effective viscosities for the three LES models that were discussed in section 5.2. These were obtained from a calculation without an LES model. (Using, e.g., a WALE model might dampen $\nu^{\text{sfs}}_{\text{WALE}}$ more than the other sub-filter scale viscosities, which would have distorted the figure.) The sub-filter terms display clear discontinuities, showing that the eddy viscosity is not just a physical model, but that it is closely linked to the discretization.

One of the most important quantities for engineering purposes is the bulk velocity, given by Eq. 5.35. Table 5.1 lists the results. For comparison, the bulk velocity can also be determined from the Darcy friction factor. From Eq. 5.38 and the definitions of the Reynolds numbers (Eqs. 5.30 and 5.34),

$$\frac{u_{\text{bulk}}}{u_\tau} = \frac{\text{Re}_{\text{bulk}}/4}{\text{Re}_\tau} = \sqrt{\frac{8}{f_{\text{D}}}} \,. \tag{5.50}$$

The bulk velocity is overestimated in all calculations except the DNS.

For further validation, figures 5.4 and 5.5 compares the averaged stresses for the model-free high-order simulation (uDNS_{32}) to the DNS data from [36] and [30], which are practically exact solutions. Our first-order statistic ($\langle u_1 \rangle$) is more accurate than the LES data (SYB), whereas the second-order statistics (the Reynolds stresses) are less accurate. Possible explanations include their dynamic LES model, and our small domain. Overall the results are satisfactory.

5.5. Discussion

This chapter has presented a large eddy simulation of turbulent plane channel flow. The results are comparable to previous literature.

Interestingly, the no-model approach (i.e., 'unresolved DNS', denoted by uDNS) yields more accurate results than the WALE model. This could be because the locally

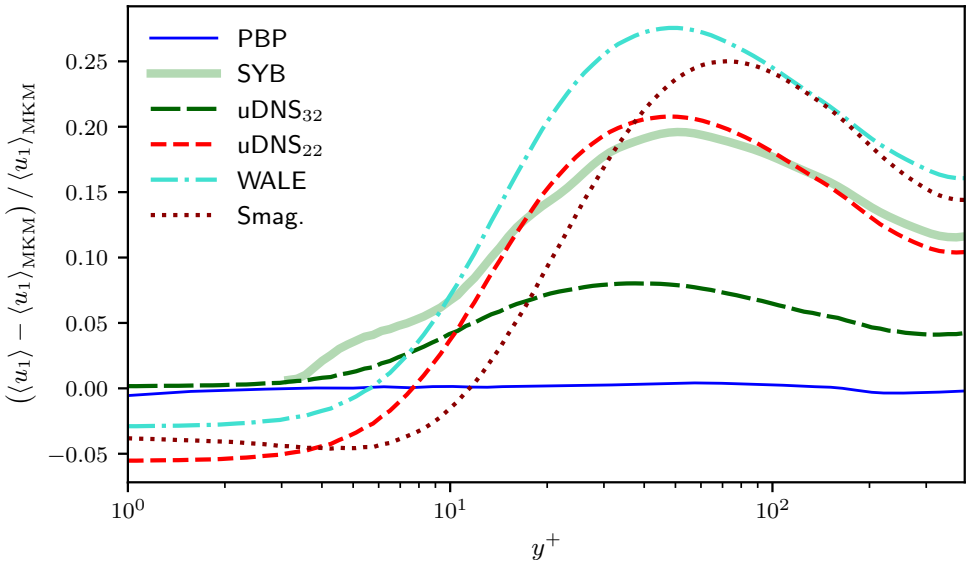


Figure 5.2: Relative deviation of $\langle u_1 \rangle$ from the DNS reference data $\langle u_1 \rangle_{\mathrm{MKM}}$ in Moser, Kim, and Mansour [36]. See Table 5.6 for the meaning of the labels.

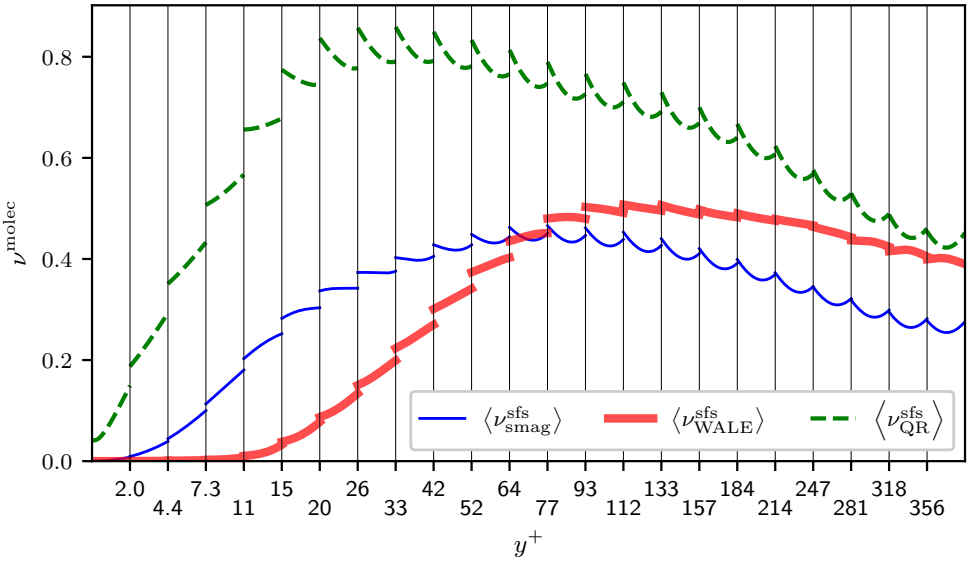


Figure 5.3: Average values of the suf-filter scale (SFS) viscosity for various LES models ($\nu^{\mathrm{sfs}}_{\mathrm{smag}}$ in Eq. 5.14, $\nu^{\mathrm{sfs}}_{\mathrm{WALE}}$ in Eq. 5.19, and $\nu^{\mathrm{sfs}}_{\mathrm{QR}}$ in Eq. 5.21), obtained from the uDNS$_{32}$ calculation (see Table 5.6). Scaled such that each element looks the same size; the vertical lines indicate the element boundaries.

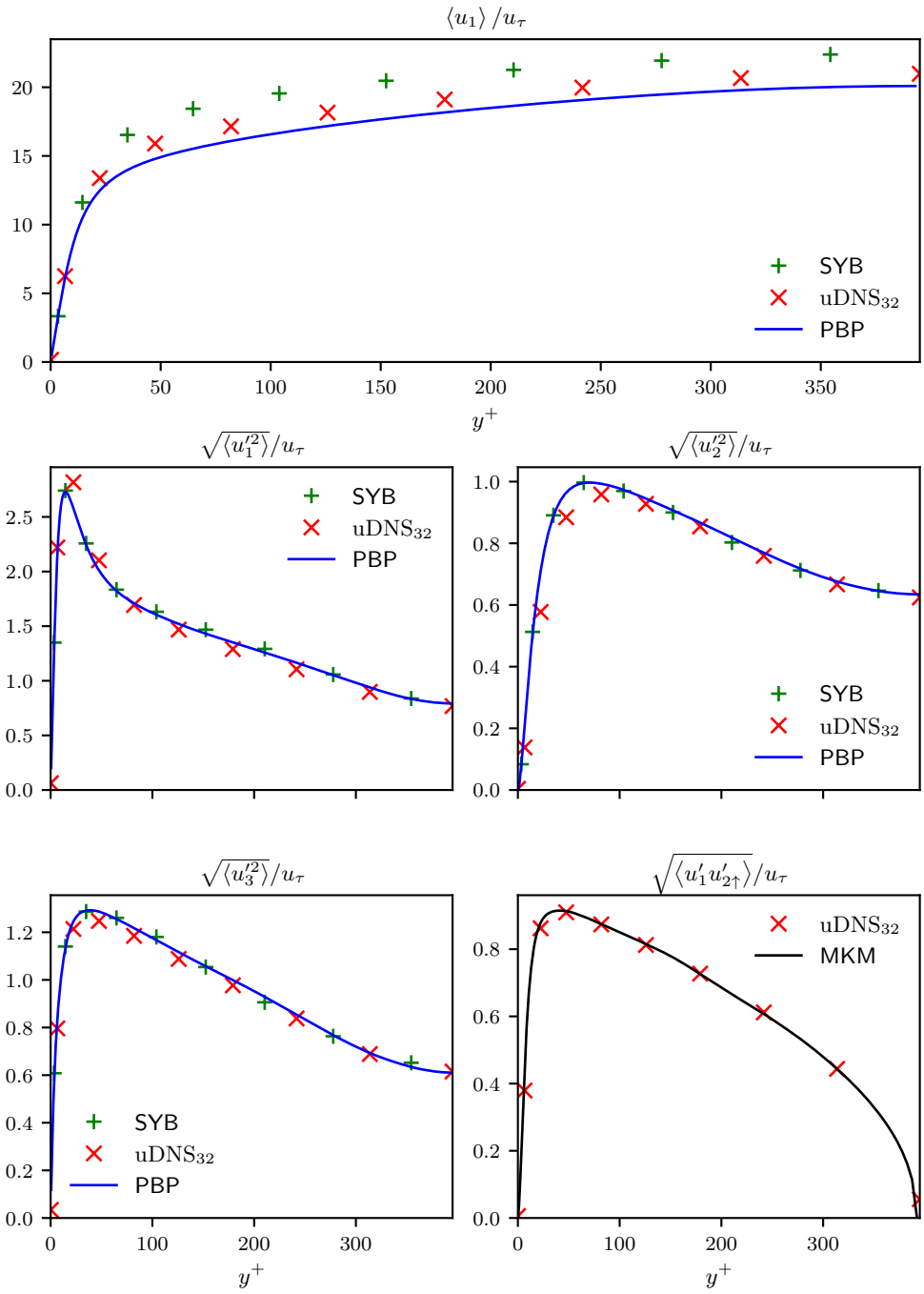


Figure 5.4: Reynolds average $\langle u_1 \rangle$ and Reynolds stress components (Eq. 5.46), compared to other LES and DNS results at $\mathrm{Re}_\tau = 395$. See Table 5.6 for the labels.

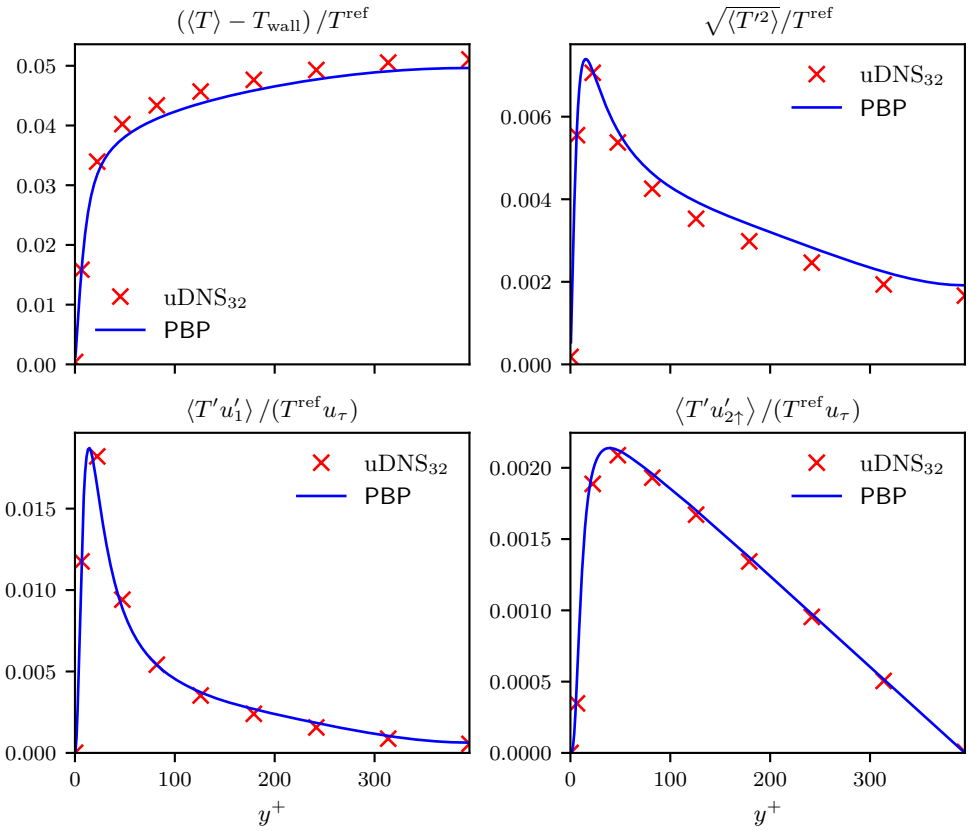


Figure 5.5: Reynolds average $\langle T\rangle$ and turbulent heat flux (Eq. 5.47), at $\mathrm{Re}_\tau = 395$, compared to DNS results. See Table 5.6 for the labels.
T^{ref} is given by Eq. 5.48. Note that $\sqrt{\langle T'^2\rangle}/T^{\text{ref}}$ equals the root-mean-square value of $(\langle T\rangle - T_{\text{wall}})/T^{\text{ref}}$.

Figure 5.6: Abbreviations used in Figs. 5.4–5.3 and Table 5.1.

PBP	DNS reference data from Patel, Boersma, and Pecnik [30], accessed through [41]
MKM	DNS reference data from Moser, Kim, and Mansour [36], accessed through [42]
SYB	LES reference data from Singh, You, and Bose [40]. Fully resolved; only contains LES model error
uDNS$_{22}$	present simulation without an LES model (i.e., unresolved DNS) with $\mathcal{P}_m = \mathcal{P}_p = \mathcal{P}_h = 2$.
WALE	same as uDNS$_{22}$, but with an WALE LES model
Smag.	same as uDNS$_{22}$, but with a Smagorinsky LES model
uDNS$_{32}$	same as uDNS$_{22}$, but with a higher order of approximation for m: $\mathcal{P}_m = 3$ and $\mathcal{P}_p = \mathcal{P}_h = 2$

Table 5.1: Bulk velocity implied by various simulations. See Table 5.6 for the labels. The Colebrook-White estimate is based on Eqs. 5.39 and 5.50.

	u_{bulk}/u_τ	deviation from MKM
MKM	17.6	
PBP	17.5	-0.10%
SYB	20.0	14%
uDNS_{22}	19.9	13%
uDNS_{32}	18.5	5.1%
WALE	21.0	20%
Smag.	20.7	18%
Colebrook-White	18.4	4.7%

conservative spatial discretization acts as an implicit LES ('iLES') model, analogous to iLES models for finite volume discretizations (e.g., [43]). Recent research indeed indicates that a DG discretization with a high-order polynomial space is an low-pass filter (e.g., [44–46]). Furthermore, note that $\langle u_1'^2 \rangle$, $\langle u_2'^2 \rangle$, and $\langle u_3'^2 \rangle$ are not equal in the bulk of the fluid, at $y = 0$, that is, $y^+ = Re_\tau = 395$, as can be seen in Fig. 5.4. The anisotropic Reynolds stress means that the effect of the walls is present in the whole fluid. It is therefore not surprising that a fixed-coefficient (i.e., non-dynamic) isotropic LES model does not model the dissipation well.

We also found that lowering the time step size degrades the solution quality. This could be explained by implicit temporal filtering due to the temporal discretization error, which might damp spurious oscillations. This can be compared to the findings by Meyers and Sagaut [47], who observed that coarser grids can sometimes result in more accurate predictions of the mean flow.

Meyers and Sagaut [47] also emphasize that, while a turbulent channel flow is easy to handle with some discretizations, it is not an easy LES test case. This is because the turbulent flow structures are generated at the walls, whereas LES models have traditionally been better at capturing the physics in isotropic flow in the bulk.

We stress that the discontinuous Galerkin method is fundamentally an unstructured discretization. It is 'blind' to any geometrical symmetries, and does not change its approximation scheme depending on the direction, as is common for finite difference in channel geometries. Infinite plane channel flow is therefore a relatively difficult test case with our discretization.

References

[1] B. Vreman, B. Geurts, and H. Kuerten, *Subgrid-modelling in LES of compressible flow,* Applied Scientific Research **54**, 191 (1995).

[2] B. Lessani, J. Ramboer, and C. Lacor, *Efficient Large-eddy Simulations of Low Mach Number Flows Using Preconditioning and Multigrid,* International Journal of Computational Fluid Dynamics **18**, 221 (2004).

[3] E. Garnier, N. Adams, and P. Sagaut, *Large Eddy Simulation for Compressible Flows*, 1st ed., Scientific Computation (Springer, Dordrecht, 2009).

[4] P. Sagaut, *Large Eddy Simulation for Incompressible Flows: An Introduction*, 3rd ed. (Springer-Verlag Berlin Heidelberg, 2006) p. 558.

[5] S. B. Pope, *Turbulent Flows*, 1st ed. (Cambridge University Press, Cambridge, UK, 2000) p. 771.

[6] L. C. Berselli, T. Iliescu, and W. J. Layton, *Mathematics of Large Eddy Simulation of Turbulent Flows*, 1st ed. (Springer, Berlin, Heidelberg, 2006).

[7] U. Piomelli, P. Moin, and J. H. Ferziger, *Model consistency in large eddy simulation of turbulent channel flows,* Physics of Fluids **31**, 1884 (1984).

[8] M. Inagaki, *A new wall-damping function for large eddy simulation employing kolmogorov velocity scale,* International Journal of Heat and Fluid Flow **32**, 26 (2011).

[9] F. Nicoud and F. Ducros, *Subgrid-Scale Stress Modelling Based on the Square of the Velocity Gradient Tensor,* Flow, Turbulence and Combustion **62**, 183 (1999).

[10] M. de la Llave Plata, V. Couaillier, and M.-C. le Pape, *On the use of a high-order discontinuous Galerkin method for DNS and LES of wall-bounded turbulence,* Computers & Fluids **176**, 320 (2018).

[11] ANSYS-Fluent, *ANSYS FLUENT 12.0/12.1 Documentation,* [Online] Accessed on 2019-01-09 (2009).

[12] R. Verstappen, *When Does Eddy Viscosity Damp Subfilter Scales Sufficiently?* Journal of Scientific Computing **49**, 94 (2011).

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

[14] A. W. Vreman, *Direct and Large-Eddy Simulation of the Compressible Turbulent Mixing Layer*, Phd thesis, University of Twente (1995).

[15] A. Abbà, L. Bonaventura, M. Nini, and M. Restelli, *Dynamic models for Large Eddy Simulation of compressible flows with a high order DG method,* Computers & Fluids **122**, 209 (2015), arXiv:1407.6591 .

[16] M. Lesieur, *Turbulence in Fluids*, 4th ed. (Springer, Dordrecht, 2008) p. 563.

[17] B. Krank, N. Fehn, W. A. Wall, and M. Kronbichler, *A high-order semi-explicit discontinuous galerkin solver for 3d incompressible flow with application to dns and les of turbulent channel flow,* Journal of Computational Physics **348**, 634 (2017).

[18] H. Nemati, *Direct numerical simulation of turbulent heat transfer to fluids at supercritical pressures*, doctoral thesis, TU Delft (2016).

[19] C. Yang, J. Xu, X. Wang, and W. Zhang, *Mixed convective flow and heat transfer of supercritical CO2 in circular tubes at various inclination angles,* International Journal of Heat and Mass Transfer **64**, 212 (2013).

[20] X. Chu and E. Laurien, *Flow stratification of supercritical CO2 in a heated horizontal pipe,* The Journal of Supercritical Fluids **116**, 172 (2016).

[21] F. Föll, S. Pandey, X. Chu, C.-D. Munz, E. Laurien, and B. Weigand, *High-Fidelity Direct Numerical Simulation of Supercritical Channel Flow Using Discontinuous Galerkin Spectral Element Method,* in *High Performance Computing in Science and Engineering ' 18* (Springer International Publishing, Cham, 2019) pp. 275–289.

[22] R. G. Rehm and H. R. Baum, *Equations of Motion for Thermally Driven, Buoyant Flows,* Journal of Research of the National Bureau of Standards **83**, 297 (1978).

[23] F. C. Nicoud, *Numerical study of a channel flow with variable properties,* in *Annual Research Briefs* (Center for Turbulence Research, 1998) pp. 289–310.

[24] R. Knikker, *A comparative study of high-order variable-property segregated algorithms for unsteady low Mach number flows,* International Journal for Numerical Methods in Fluids **66**, 403 (2011).

[25] M. Avila, R. Codina, and J. Principe, *Large eddy simulation of low Mach number flows using dynamic and orthogonal subgrid scales,* Computers & Fluids **99**, 44 (2014).

[26] G. Accary, P. Bontoux, and B. Zappoli, *Turbulent Rayleigh-Bénard convection in a near-critical fluid by three-dimensional direct numerical simulation,* Journal of Fluid Mechanics **619**, 127 (2009).

[27] G. Accary and I. Raspo, *A 3D finite volume method for the prediction of a supercritical fluid buoyant flow in a differentially heated cavity,* Computers & Fluids **35**, 1316 (2006).

[28] W. Wang and S. He, *Direct numerical simulation of fluid flow at supercritical pressure in a vertical channel,* in *International Topical Meeting on Nuclear Reactor Thermal Hydraulics 2015, NURETH 2015*, Vol. 3 (Chicago, USA, 2015) pp. 2334–2347.

[29] J. W. R. Peeters, R. Pecnik, M. Rohde, T. H. J. J. van der Hagen, and B. J. Boersma, *Turbulence attenuation in simultaneously heated and cooled annular flows at supercritical pressure,* Journal of Fluid Mechanics **799**, 505 (2016).

[30] A. Patel, B. J. Boersma, and R. Pecnik, *Scalar statistics in variable property turbulent channel flows,* Physical Review Fluids **2**, 084604 (2017).

[31] R. B. Bird, W. E. Stewart, and E. N. Lightfoot, *Transport Phenomena*, 2nd ed., edited by P. Kulek (John Wiley & Sons, Inc., New York, NY, 2002) p. 914.

[32] D. J. Zigrang and N. D. Sylvester, *Explicit approximations to the solution of Colebrook's friction factor equation,* AIChE Journal **28**, 514 (1982).

[33] J. Eggels, *Direct and Large Eddy Simulation of Turbulent Flow in a Cylindrical Pipe Geometry*, doctoral thesis, Delft University of Technology (1994).

[34] J. Kühnen, B. Song, D. Scarselli, N. B. Budanur, M. Riedl, A. P. Willis, M. Avila, and B. Hof, *Destabilizing turbulence in pipe flow,* Nature Physics **14**, 386 (2018).

[35] H. Abe, H. Kawamura, and Y. Matsuo, *Surface heat-flux fluctuations in a turbulent channel flow up to Re_τ=1020 with Pr=0.025 and 0.71,* International Journal of Heat and Fluid Flow **25**, 404 (2004).

[36] R. D. Moser, J. Kim, and N. N. Mansour, *Direct numerical simulation of turbulent channel flow up to $Re\tau$=590,* Physics of Fluids **11**, 943 (1999).

[37] U. Piomelli, C. Benocci, and J. P. A. J. Van Beek, *Large Eddy Simulation: Theory and Simulation,* (2016), lectures notes.

[38] C. C. de Wiart, K. Hillewaert, L. Bricteux, and G. Winckelmans, *Implicit LES of free and wall-bounded turbulent flows based on the discontinuous Galerkin/symmetric interior penalty method,* International Journal for Numerical Methods in Fluids **78**, 335 (2015).

[39] S. S. Collis, *The DG/VMS Method for Unified Turbulence Simulation,* in *32nd AIAA Fluid Dynamics Conference and Exhibit* (Houston, Texas, 2002).

[40] S. Singh, D. You, and S. T. Bose, *Large-eddy simulation of turbulent channel flow using explicit filtering and dynamic mixed models,* Physics of Fluids **24**, 85105 (2012).

[41] R. Pecnik, *DNS data from variable property channel flows,* Github. [Online] Accessed on 2019-10-09 (2019).

[42] R. D. Moser, J. Kim, and N. N. Mansour, *DNS Data for Turbulent Channel Flow,* University of Texas. [Online] Accessed on 2021-05-02 (2007).

[43] S. Hickel, A. Devesa, and N. A. Adams, *Implicit Turbulence Modeling by Finite Volume Methods,* in *Numerical Simulation of Turbulent Flows and Noise Generation: Results of the DFG/CNRS Research Groups FOR 507 and FOR 508*, edited by C. Brun, D. Juvé, M. Manhart, and C.-D. Munz (Springer Berlin Heidelberg, Berlin, Heidelberg, 2009) pp. 149–173.

[44] E. Ferrer, *An interior penalty stabilised incompressible discontinuous Galerkin-Fourier solver for implicit large eddy simulations,* Journal of Computational Physics **348**, 754 (2017).

[45] R. C. Moura, G. Mengaldo, J. Peiró, and S. J. Sherwin, *On the eddy-resolving capability of high-order discontinuous Galerkin approaches to implicit LES / under-resolved DNS of Euler turbulence,* Journal of Computational Physics **330** (2017), 10.1016/j.jcp.2016.10.056.

[46] A. R. Winters, R. C. Moura, G. Mengaldo, G. J. Gassner, S. Walch, J. Peiro, and S. J. Sherwin, *A comparative study on polynomial dealiasing and split form discontinuous Galerkin schemes for under-resolved turbulence computations,* Journal of Computational Physics **372**, 1 (2018), arXiv:1711.10180 .

[47] J. Meyers and P. Sagaut, *Is plane-channel flow a friendly case for the testing of large-eddy simulation subgrid-scale models?* Physics of Fluids **19**, 48105 (2007).

6

Conclusion

The previous chapters have treated various discretizations in computational fluid dynamics without an overarching theme. The conclusions and their relevance are therefore best described by considering the chapters separately.

Chapter 2 has explained our discontinuous Galerkin method, which differs from most other literature on two minor points. First, the penalty parameters (for the interior penalty and Lax-Friedrichs numerical fluxes) are evaluated in a pointwise manner, rather than being averaged over the face, or a neighboring element. Second, the convection at Dirichlet boundaries is treated correctly, which is not the case in most of the listed references. In practice these points might not make much difference to the numerical outcomes.

The pressure correction method in chapter 3 is almost completely standard. The one peculiarity is that the projection step (Eq. 3.10) contains a penalty matrix, which is a consequence of using an equal-order finite element discretization. As explained in section 3.5.1, this incurs a substantial computational cost at high Reynolds numbers and small time steps, though this problem could be particular to our implementation and solver.

Chapter 4 treats the variable density in great detail. We pay particular attention to the most convenient form of the enthalpy equation (primitive or conservative) and whether to solve for the specific enthalpy or the volumetric enthalpy. Each approach has its own challenges; it is not completely clear which is best. We decided to solve for the specific enthalpy from the conservative form of the transport equation.

The biggest potential problem with this choice is that it complicates the temporal discretization of the enthalpy, which can destabilize the calculation, or degrade the order of convergence. It was shown that these problems can be addressed by a simple modification of the finite difference scheme ('method #2'), and by 'offsetting' the specific enthalpy with a constant h_0. The choice of this enthalpy offset was central to stabilizing the coupled transport equations with a temperature-dependent density.

The value of h_0 would also have been important if we had solved for the volumetric enthalpy (as explained in section 4.1.2), because it determines whether there is a one-to-one relation between the volumetric enthalpy and the density. It is therefore surprising that we have not seen this mentioned previous literature. Hopefully the discussions in chapter 4 are useful to the development of other numerical methods for low-Mach number flows.

Chapter 5 features turbulent flow simulations between two infinite planes. This geometry was chosen for the abundance of high-quality reference data and its periodic boundary conditions, which reduce the computational cost, because there are fewer walls that require grid refinement. The discontinuous Galerkin method and our solver are fundamentally unstructured, and do not exploit the simplicity of the geometry. The test case is therefore no less challenging than if the mesh were ostensibly more complicated. Our LES models have also been used by others with similar discretizations, so our results add little to the existing body of literature, though they can be considered further validation of the spatial and temporal discretizations in chapters 2–3 and their implementation.

Acknowledgements

Sections 2.1.2–2.4, 2.7, 3.1–3.2, and 4.1–4.4 are largely based on work that I did at TU Delft, where most of the code `DGFlows` was also written. I am grateful to Danny Lathouwers for having meant to help with the scientific aspect of my work, and I am happy that he has always kept his sense of humor. Danny suggested method #2 as an improvement over method #1 in section 4.3. I would like to thank Marco Tiberga for having been a pleasant colleague, and for his help with the code development. Among other things, Marco wrote the initial implementation of the penalty terms in Eq. 5.26.

My employment as a PhD candidate at TU Delft was funded by the sCO_2-HeRo project that has received funding from the European research and training program 2014–2018 (grant agreement ID 662116).

I am grateful to prof. dr. Laurien Eckart and prof. dr. Jörg Starflinger for having given me the opportunity to work at the Institut für Kernenergetik und Energiesysteme (IKE) at the University of Stuttgart, without which this thesis would not have been realized. I was graciously given access to the HazelHen supercomputer at the High-Performance Computing Center Stuttgart (HLRS), which greatly sped up the development and implementation of the large eddy simulation models. Sandeep Pandey contributed significantly with practical tips on the turbulent channel flow simulations.

My stay at IKE in Stuttgart was partially supported by the ENEN+ project that has received funding from the Euratom research and training Work Programme 2016–2017 — 1 #755576.

I would like to thank Rene Pecnik for the discussion on the discretization of the enthalpy equation with a variable density. Rene also provided access to the Reynolds computer cluster at the TU Delft, which was used for some of the calculations in section 5.4.

Ten slotte ben ik erg gesteund door het geduld en meeleven van Matteo. Ik ben je onbeschrijvelijk dankbaar voor het vertrouwen, het lachen, de herinneringen en de liefde die je me de afgelopen jaren gegeven hebt en zie ernaar uit om elkaars levens daar nog lang mee te vullen.

Curriculum Vitæ

Aldo Hennink

1990	Born in Amsterdam, The Netherlands.

Education

2012	Msc. in Physics Technical University of Delft
2015	Msc. in Physics Technical University of Delft
2021	PhD. in Physics Technical University of Delft *Thesis:* Low-Mach Number Flow and the Discontinuous Galerkin Method *Promotors:* dr. D. Lathouwers Prof. dr. J. L. Kloosterman

List of Publications

3. **A. Hennink**, M. Tiberga, D. Lathouwers, *A Pressure-based solver for low-Mach number flow using a discontinuous Galerkin method*, Journal of Computational Physics, **425**, 109877 (2021).

2. M. Tiberga, **A. Hennink**, J. L. Kloosterman, D. Lathouwers, *A high-order discontinuous Galerkin solver for the incompressible RANS equations coupled to the k-ϵ turbulence model*, Computers & Fluids, **212**, 104710 (2020).

1. **A. Hennink**, D. Lathouwers, *A discontinuous Galerkin method for the mono-energetic Fokker-Planck equation based on a spherical interior penalty formulation*, Journal of Computational and Applied Mathematics **330**, 253-267 (2018).