Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

MLMC-FD method for 2D statistical solutions of the Navier-Stokes equations

Yann Poltera
MSc. Thesis in Computational Science
ETH Zürich

supervised by
Prof. Dr. Ch. Schwab, Prof. Dr. L. Kleiser
Mathematics, Mechanical and Process Engineering
ETH Zürich

Abstract

The multilevel Monte Carlo (MLMC) Finite Difference (FD) simulation of statistical solutions of the (incompressible) Navier-Stokes equations (NSE) as described in [2] is proposed. The corresponding probability measure μ_t on the ensemble of Leray solutions of the NSE is approximated by sample averages on a hierarchic family of discretizations in space and time. Uniform measures μ_0 are considered. Direct numerical simulations of NSE for the pathwise solutions are performed, using the code IMPACT of Kleiser et al. [6]. The effect of under-resolved scales for coarse grid samples in the MLMC-FD on the overall accuracy is investigated. Efficient parallelization and a load balancing strategy of the MLMC algorithm on distributed memory architectures are proposed along the lines of [17]. Numerical results in two spatial dimensions, with periodic boundary conditions on large scale, parallel computers are presented.

Information

Title: MLMC-FD method for 2D statistical solutions of the Navier-Stokes equations
Author: Yann Poltera, ypoltera@student.ethz.ch
Supervisor: Prof. Dr. Ch. Schwab, Prof. Dr. L. Kleiser

Mathematics, Mechanical and Process Engineering
ETH Zürich
Rämistrasse 101, 8092 Zürich
www.ethz.ch

Acknowledgements

I would like to thank Prof. Dr. Christoph Schwab and Prof. Dr. Leonhard Kleiser for letting me write this thesis under their supervision. A grateful thank goes to the supervising assistants Dr. Andrea Barth and Tarun Chadha, and to Jonas Šukys for his technical support.

I thank also the team of the Swiss National Supercomputing Center, CSCS, at Lugano [13], for providing support and computational resources under the project ID 'g54'.

Contents

List of Figures

List of Tables

Introduction

The incompressible Navier-Stokes equations (NSE) govern the motion of constant property Newtonian fluids. These are deterministic non-linear equations, however, at high Reynolds numbers, their solution display a chaotic, or turbulent, behavior, and are very sensitive to small perturbations in initial conditions, boundary conditions and material properties [14, sect. 3.1]. It is therefore accepted that turbulent flows are statistical in nature [2, sect. 5.0].

A statistical solution describes the evolution of the probability distribution of a random variable that satisfies some dynamical behavior. In the context of statistical solutions of the Navier-Stokes equations, we assume we are given an initial probability distribution on the ensemble of all physically meaningful velocity fields, and consider the evolution of the probability distribution in time as the initial velocities evolve. In particular, we consider the evolution of statistical moments of the probability distribution. Such moments (which are ensemble averages of some quantity of interest) are of importance in a variety of contexts.

In [1], a novel theory and computational approach to compute generalized moments of statistical solutions of the incompressible Navier-Stokes equations has been presented. The approach consists in a multilevel Monte Carlo sampling strategy combined with the use of space and time discretization methods for each sample. It permits to capture efficiently ensemble averages and bulk properties of viscous, incompressible flows, because it can compensate to some extent under-resolved discretizations by statistical oversampling [1, sect. 9]. In this thesis we study the theoretical concepts that lead to this approach, and test the method for two-dimensional incompressible laminar flows with periodic boundary conditions, on large-scale, parallel computers. The thesis is structured as follows.

In Chapter 1, we review the fluid dynamics model behind the incompressible Navier-Stokes equations, starting from the continuum hypothesis and continuing towards the formulation of conservation laws of continuum mechanics applied to fluids. The assumption of constant material properties will then lead us to the incompressible Navier-Stokes equations. Then, following the description in [2], we present concepts and results from the mathematical theory of the Navier-Stokes equations. In particular, we introduce the solution spaces of finite kinetic energy and finite enstrophy for no-slip and periodic boundary conditions with associated norms, the Stokes operator, whose eigenfunctions constitute an orthonormal basis of the solution spaces, the functional formulation and the weak formulation of the Navier-Stokes equations, whose solutions are called Leray solutions. We conclude the chapter by presenting explicitly an orthonormal basis for the space of divergence-free periodic velocity fields, which we will use to expand data in our numerical experiments.

In Chapter 2, further following the presentation in [2], we turn to statistical solutions, which are one-parameter family of probability measures that satisfy an evolution equation, starting from a given initial probability measure on an ensemble of initial velocities. We introduce first the concept of (generalized) statistical moments, present then an evolution equation for these statistical moments, which will lead to the definition of statistical solutions of the Navier-Stokes equations, and conclude with an existence and uniqueness result.

In Chapter 3, we present the Monte Carlo (MC) sampling strategy, from which ensemble averages and bulk properties of the statistical solution can be approximated by sampling from the initial probability distribution and calculating a sample mean. We then present a result from [1], that states that, with mild assumptions, the error (in a statistical sense) of the sample mean decreases with the square root of the number of samples, and this independently of the kinematic viscosity.

In Chapter 4, following the presentation and results in [1], we discuss the effect of using space and time discretization methods to approximate the sample solutions used in the Monte Carlo approach. This results in the singlelevel Monte Carlo method (MC). We then present the multilevel Monte Carlo (MLMC) method, in which the statistical moments are approximated numerically by sample averages on a hierarchic family of discretizations in space and time, and permits to equilibrate statistical and discretization errors more efficiently than the singlelevel Monte Carlo method.

In Chapter 5, we give a detailed description of the solver that we use in our numerical experiments to approximate the sample solutions in the MC and MLMC methods. The solver is named 'IMPACT' [6] and is a massively parallel solver for incompressible flows which uses Finite Differences (FD) in both space and time for the discretization and solves the resulting linear systems iteratively.

In Chapter 6, we present the MLMC-FD solver developed in the context of this thesis, which uses the IMPACT code to calculate the pathwise evolutions of randomly generated initial velocity samples that are used in the MLMC method. This solver was implemented on top of the IMPACT solver for a usage on parallel computers, based on a static load balancing strategy presented in [17].

Finally, in Chapter 7, we present results from numerical experiments on large-scale, parallel computers, where two-dimensional incompressible flows with periodic boundary conditions and uniform probability distributions for the generation of the initial data were considered. In these experiments, the MLMC-FD solver is first tested. Then, further tests are done, where the convergence of the error of the MLMC method is measured and compared with the theoretical predictions, and where the effect of under-resolved scales on coarse grid samples on the overall accuracy of the approximated statistical moments is investigated.

Chapter 1

Fluid dynamics model and elements of the mathematical theory of the Navier-Stokes equations

In the first part of this chapter, we present the fluid dynamics model behind the Navier-Stokes equations, following the description in [11], [14] and [2]. A detailed description can also be found in [12]. We first briefly review the physical principles behind the continuous representation of fluids, and present two different continuous representations. Then we review the basic conservation principles of continuum mechanics applied to fluids, which will lead to the incompressible Navier-Stokes equations that govern the flow of constant-property Newtonian fluids.

In the second part of this chapter, we present concepts and results from the mathematical theory of the incompressible Navier-Stokes equations that can be found in the book of Foias et al. [2], as these are needed to define statistical solutions of the Navier-Stokes equations.

1.1 Continuum hypothesis and continuous representations

As mentioned in [14, chapt. 2.1], the "idea of treating fluids as continuous media is both natural and familiar", and simplifies the physical modeling of the dynamical behavior of fluids [11].

We review now the continuum hypothesis.

1.1.1 Continuum hypothesis

In the continuum hypothesis, a fluid is considered as a continuum that is abstracted from the underlying molecular structure. Instead of considering the detailed molecular structure, it is assumed that a fluid consists of a dense packing of individual elements, so-called *fluid particles*, that occupy the space continuously. These fluid particles are small compared to the relevant scales of the flow, but large compared to the molecular scales [11]. The

separation of length scales is quantified by the Knudsen number

$$\mathrm{Kn} = \frac{\lambda}{l} \,, \tag{1.1.1}$$

where $\lambda \in \mathbb{R}_{>0}$ represents the molecular collision mean free path and $l \in \mathbb{R}_{>0}$ the smallest representative physical scale in a flow. In general the continuum approach, adopted throughout this thesis, is "appropriate for $\mathrm{Kn} \ll 1$" [14, chapt. 2.1]. There, fluid particles and points in space are mapped one-to-one, i.e. at every point in space there is exactly one fluid particle and every fluid particle is located at a unique point in space. Then, physical properties of a fluid (such as the velocity or the density) can be represented at every point in space (and, as such, for every fluid particle) by a continuous field quantity [11]. This quantity is defined as an average over a small spherical region whose radius is small compared to l but large compared to λ [14, chapt. 2.1]. It is "important to appreciate that, once we invoke the continuum hypothesis to obtain continuous fields, we can leave behind all notions of the discrete molecular nature of the fluid, and molecular scales cease to be relevant" [14, chapt. 2.1].

1.1.2 Lagrangian and Eulerian representations

Since fluids are treated as continuous media in the continuum hypothesis, they need to obey the basic conservation principles of continuum mechanics: conservation of mass, linear momentum and energy [2, chapt. 1.1]. The corresponding equations can be written according to two different representations: the Eulerian and the Lagrangian representation.

Eulerian representation

Consider the velocity field

$$\mathbf{u} : \begin{cases} D \times \bar{J} \to \mathbb{R}^d \\ (\mathbf{x}, t) \mapsto \mathbf{u}(\mathbf{x}, t) \end{cases} \tag{1.1.2}$$

on the bounded and connected domain $D \subset \mathbb{R}^d$, in space dimension $d = 2$ or 3, and on the finite time interval $\bar{J} = [0, T]$, with $T < \infty$.
The velocity field $\mathbf{u}(\mathbf{x}, t)$ represents the fluid velocity at point $\mathbf{x}$ at time t as seen from an inertial reference frame. Other fields such a the density field $\rho(\mathbf{x}, t) : D \times \bar{J} \mapsto \mathbb{R}_{>0}$ or the hydrostatic pressure field $p(\mathbf{x}, t) : D \times \bar{J} \mapsto \mathbb{R}$ are defined similarly.

This is the so-called *Eulerian* representation [14, chapt. 2.2]. It is the representation that we will use to formulate the conservation laws.

Lagrangian representation

Another useful represention is the *Lagrangian* representation, where quantities of interest of a moving fluid particle at a fixed, specific time are given with respect to the particle's initial position [14, chapt. 2.2].

In other words, $\mathbf{u}(\mathbf{x}_0; t)$ is the velocity at time t of a fluid particle that was located at the point $\mathbf{x}_0$ at time 0. It is the representation that we will use to derive the conservation laws.

Both representations can be related through

$$\mathbf{u}(\mathbf{x}_0; t) = \mathbf{u}(\mathbf{x}(\mathbf{x}_0; t), t) \,, \tag{1.1.3}$$

where $\mathbf{x}(\mathbf{x}_0; t)$ is the position at time t of a fluid particle that was located at the point $\mathbf{x}_0$ at time 0. The Lagrangian representations of other quantities such as the density $\rho(\mathbf{x}_0; t)$

or the hydrostatic pressure $p(\mathbf{x}_0;t)$ are defined similarly.

We can derive $\mathbf{x}(\mathbf{x}_0;t)$ by solving the initial value problem

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y}(t) = \mathbf{u}(\mathbf{y}(t),t) \quad , \quad \mathbf{y}(0) = \mathbf{x}_0 \quad , \quad \mathbf{y}(t) = \mathbf{x}(\mathbf{x}_0;t)\ . \tag{1.1.4}$$

Its solution $t \to \mathbf{y}(t)$ defines the path travelled by a particle carried along by the fluid, a particle trajectory, also called a *pathline*.
A *streamline* at time τ is the curve in D defined by the autonomous initial value problem

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y}(t) = \mathbf{u}(\mathbf{y}(t),\tau) \quad , \quad \mathbf{y}(0) = \mathbf{y}_0\ . \tag{1.1.5}$$

Remark that streamlines are in general not the same as pathlines, except if the velocity field is stationary [11].

Remark 1.1.1. A velocity field induces a transformation (mapping) of space [7]. Consider the path of a particle located at $\mathbf{x}_0 \in D$ at time $t = 0$, and assume that the particle does not leave the domain between time $t = 0$ and time $t = \tau > 0$ (this assumption is not necessary if $\mathbf{u}\cdot\mathbf{n}|_{\partial D} = 0$ (where $\mathbf{n}$ is the outward unit normal vector to the boundary ∂D) or if $\mathbf{u}$ is periodic). Then the mapping

$$\Phi^\tau : \begin{cases} D \to D \\ \mathbf{x}_0 \mapsto \mathbf{x}(\mathbf{x}_0;\tau) \end{cases} \quad , \quad \tau \mapsto \mathbf{x}(\mathbf{x}_0;\tau) \text{ solution of IVP (1.1.4)} \tag{1.1.6}$$

is a well defined mapping of D to itself, and is called the *flow map*. Obviously, it satisfies

$$\Phi^0\mathbf{x}_0 = \mathbf{x}_0\ . \tag{1.1.7}$$

Moreover, $V = \Phi^\tau(V_0)$ is the volume occupied at time $t = \tau$ by particles that occupied $V_0 \subset D$ at time $t = 0$, assuming τ is small enough such that none of the particles in V_0 have left D between time $t = 0$ and time $t = \tau$.

1.1.3 Material derivative and Reynolds transport theorem

Consider some fluid property $\phi : D \times \bar{J} \to \mathbb{R}$ of a fluid particle that is located at point $\mathbf{x}$ at time t and that is moving with the flow described by the velocity field $\mathbf{u}$. The material derivative of ϕ is defined as

$$\frac{\mathcal{D}}{\mathcal{D}t}\phi := \left.\frac{\mathrm{d}}{\mathrm{d}t}\phi\right|_{(\mathbf{x}=\mathbf{x}(t),t)} = \frac{\partial}{\partial t}\phi + \mathbf{u}\cdot\nabla\phi \tag{1.1.8}$$

and takes into account both local and advective parts of the total derivative in time. It depicts the rate of change as experienced from the moving particle [11].

Consider now the integral of ϕ over a volume V at time t that is moving with the flow, i.e.

$$\Psi = \int_V \phi\, dV\ . \tag{1.1.9}$$

The material derivative of Ψ is defined as

$$\frac{\mathcal{D}}{\mathcal{D}t}\Psi := \frac{\mathrm{d}}{\mathrm{d}t}\Psi = \frac{\mathrm{d}}{\mathrm{d}t}\int_{V=V(t)} \phi\, dV = \int_V \frac{\partial}{\partial t}\phi\, dV + \int_{\partial V} \phi\mathbf{u}\cdot\mathbf{n}\, dS\ , \tag{1.1.10}$$

where $S = \partial V$ is the surface of the volume V. It depicts the rate of change as experienced from the moving volume. The last equality is known as the Reynolds transport theorem [11].

We remark that the material derivative does not commute in general with integration. Indeed, we have

$$\frac{\mathcal{D}}{\mathcal{D}t}\int_V \phi \, dV = \int_V \frac{\mathcal{D}}{\mathcal{D}t}\phi \, dV + \int_V \phi \mathrm{div}(\mathbf{u}) \, dV \; . \tag{1.1.11}$$

We are now ready to formulate the conservation laws.

1.2 Conservation laws

For the derivations made in this section, we refer to [11] if not specified otherwise. A detailed description can also be found in [12, chapt. 4].

1.2.1 Conservation of mass

Let us denote by m the total mass of a moving volume V, i.e.

$$m = \int_V \rho \, dV \; , \tag{1.2.1}$$

where $\rho : D \times \bar{J} \mapsto \mathbb{R}_{>0}$ is the density. By definition, the mass of the volume is conserved [11] (supposing that we are not in the relativistic velocity regime), i.e.

$$\frac{\mathcal{D}}{\mathcal{D}t} m = 0 \; . \tag{1.2.2}$$

With (1.1.10), we may rewrite this in the form of a conservation law:

$$\begin{aligned} &\text{for any control volume } V \subset D : \\ &\underbrace{\int_V \frac{\partial}{\partial t}\rho \, dV}_{\text{mass change inside the volume}} + \underbrace{\int_{\partial V} \rho \mathbf{u} \cdot \mathbf{n} \, dS}_{\text{mass flux through the surface}} = 0 \; . \end{aligned} \tag{1.2.3}$$

This leads to the following partial differential equation, also called *continuity* equation:

$$\frac{\partial \rho}{\partial t} + \mathrm{div}(\rho \mathbf{u}) = 0 \; . \tag{1.2.4}$$

We have: $[m] = \mathrm{kg}$, $[\rho] = \frac{\mathrm{kg}}{\mathrm{m}^3}$, $[\mathbf{u}] = \frac{\mathrm{m}}{\mathrm{s}}$, $[\mathbf{x}] = \mathrm{m}$, $[t] = \mathrm{s}$.

Incompressible flow

A flow is called *incompressible* if

$$\frac{1}{\rho}\frac{\mathcal{D}\rho}{\mathcal{D}t} = \frac{1}{\rho}(\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho) = 0 \; . \tag{1.2.5}$$

For such flows, the continuity equation (1.2.4) reduces to

$$\mathrm{div}(\mathbf{u}) = 0 \; . \tag{1.2.6}$$

A special case of incompressible flows are flows of *incompressible*, *homogeneous* fluids, for which $\rho = const$ [11].

Remark 1.2.1. An incompressible flow has the property that it's associated flow map is volume preserving [7], i.e.

$$|\Phi^\tau(V)| = |\Phi^0(V)| \tag{1.2.7}$$

for all sufficiently small times $\tau > 0$ and for all control volumes $V \subset D$.

1.2.2 Conservation of momentum

Let us denote by $\mathbf{P}$ the total linear momentum of a moving volume V, i.e.

$$\mathbf{P} = \int_V \rho \mathbf{u}\, dV \ . \tag{1.2.8}$$

By Newton's second law of motion [14, chapt. 2.4], the rate of change of momentum experienced by the volume is equal to the sum of all forces $\mathbf{F}_{tot}$ acting on the volume:

$$\frac{\mathcal{D}}{\mathcal{D}t}\mathbf{P} = \mathbf{F}_{tot} \ . \tag{1.2.9}$$

Fluid volume elements can experience two kinds of forces: surface forces and body/volume forces [14, chapt. 2.4].

Surface forces

A fluid is differentiated from another material (e.g. a solid) by the property that, at rest, it only take pressure forces (resulting from a compressional stress) without entering in motion. Other shear and tensional forces put the fluid in motion, and these forces result from viscous stresses (that appear when the fluid is 'being deformed'). In comparison, a solid can support shear or tensional forces without entering in motion. These forces result from elastic stresses (that appear when the solid is 'being held in a static deformed configuration'), to which by the above definition a fluid material cannot be exposed [11].

These stresses are in general described by the symmetric Cauchy stress tensor $\boldsymbol{\sigma} : D \times \bar{J} \to \mathbb{R}^{d\times d}$, which takes the form

$$\sigma_{ij} = -p\delta_{ij} + \tau_{ij} \ , \tag{1.2.10}$$

where $\boldsymbol{\tau} : D \times \bar{J} \to \mathbb{R}^{d\times d}$ is the viscous stress tensor and p is the hydrostatic pressure [11]. The force on the surface ∂V of a fluid volume element V takes then the form

$$\mathbf{F}_{surface} = \int_{\partial V} \boldsymbol{\sigma} \cdot \mathbf{n}\, dS \ . \tag{1.2.11}$$

Viscous stresses are due to the molecular exchange of momentum between neighboring fluid layers with a non-zero velocity gradient. They have a dissipative effect and can therefore be seen as 'friction' terms [11].

In this report we only consider Newtonian fluids, for which it is assumed that the viscous stress tensor depends linearly on the velocity gradients and where the Cauchy stress tensor reads [14, chapt. 2.8]

$$\sigma_{ij} = -p\delta_{ij} + \underbrace{2\mu S_{ij} + \mu' \mathrm{div}(\mathbf{u})\delta_{ij}}_{\tau_{ij}} \ , \tag{1.2.12}$$

where

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{1}{3}\mathrm{div}(\mathbf{u})\delta_{ij} \tag{1.2.13}$$

is the rate-of-strain tensor,

$$\operatorname{div}(\mathbf{u})\delta_{ij} \tag{1.2.14}$$

is the dilatation tensor,

$$\mu > 0 \tag{1.2.15}$$

is the shear viscosity coefficient, called also *dynamic* viscosity, and

$$\mu' > 0 \tag{1.2.16}$$

is the dilation viscosity coefficient. The viscosity coefficients depend in general on the temperature T of the fluid [11].
We have: $[\boldsymbol{\sigma}] = \frac{\mathrm{N}}{\mathrm{m}^2}$, $[\mu] = [\mu'] = \frac{\mathrm{Ns}}{\mathrm{m}^2}$, $[p] = \frac{\mathrm{N}}{\mathrm{m}^2}$, $[T] = \mathrm{K}$.

Body forces

Body forces act on the entire fluid volume element V, and take the form

$$\mathbf{F}_{body} = \int_V \rho\mathbf{f}\, dV + \mathbf{F}_{ext}\ , \tag{1.2.17}$$

where $\mathbf{f} : D \times \bar{J} \to \mathbb{R}^d$ is a body/volume force and $\mathbf{F}_{ext} : \bar{J} \to \mathbb{R}^d$ is an external force that appears in some cases, which does not influence the fluid's motion. For example, we have an external force to hold a pipe in place [11].
We have: $[\mathbf{f}] = \frac{\mathrm{m}}{\mathrm{s}^2}$, $[\mathbf{F}_{ext}] = \mathrm{N}$.

Examples of body forces are given by

- Gravitation, with $\mathbf{f} = -g\mathbf{e}_3$.
- Forces $\mathbf{f} = \mathbf{f}_{\text{n.i.}}$ appearing when the reference frame is a non-inertial frame, e.g. Coriolis effect in the atmosphere due to the rotation of the earth [14, chapt. 2.9].

We have then

$$\begin{aligned}\frac{\mathcal{D}}{\mathcal{D}t}\mathbf{P} &= \mathbf{F}_{tot}\\ &= \mathbf{F}_{surface} + \mathbf{F}_{body}\\ &= \int_{\partial V} \boldsymbol{\sigma}\cdot\mathbf{n}\, dS + \int_V \rho\mathbf{f}\, dV + \mathbf{F}_{ext}\\ &= -\int_{\partial V} p\mathbf{n}\, dS + \int_{\partial V} \boldsymbol{\tau}\cdot\mathbf{n}\, dS + \int_V \rho\mathbf{f}\, dV + \mathbf{F}_{ext}\ .\end{aligned} \tag{1.2.18}$$

With (1.1.10), we may rewrite this in the form of a conservation law:

$$\begin{aligned}&\text{for any control volume } V \subset D:\\ &\underbrace{\int_V \frac{\partial}{\partial t}(\rho\mathbf{u})\, dV}_{\text{momentum change inside the volume}} + \underbrace{\int_{\partial V} (\rho\mathbf{u})\mathbf{u}\cdot\mathbf{n}\, dS}_{\text{momentum flux through the surface}} =\\ &\underbrace{-\int_{\partial V} p\mathbf{n}\, dS}_{\text{pressure forces}} + \underbrace{\int_{\partial V} \boldsymbol{\tau}\cdot\mathbf{n}\, dS}_{\text{viscous forces}} + \underbrace{\int_V \rho\mathbf{f}\, dV}_{\text{body forces}} + \underbrace{\mathbf{F}_{ext}}_{\text{external forces}}\ .\end{aligned} \tag{1.2.19}$$

This leads to the following partial differential equation, also called *momentum* equation:

$$\frac{\partial}{\partial t}(\rho\mathbf{u}) + \operatorname{div}((\rho\mathbf{u})\mathbf{u}) = -\nabla p + \operatorname{div}(\boldsymbol{\tau}) + \rho\mathbf{f}\ . \tag{1.2.20}$$

Inserting the continuity equation (1.2.4) into the momentum equation leads to

$$\rho\big(\frac{\partial}{\partial t}\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u}\big) = -\nabla p + \mathrm{div}(\boldsymbol{\tau}) + \rho\mathbf{f}\ . \tag{1.2.21}$$

We can rewrite (1.2.21) using Einstein's summation convention and get

$$\rho\big(\frac{\partial}{\partial t}u_i + u_j\frac{\partial u_i}{\partial x_j}\big) = -\frac{\partial}{\partial x_i}p + \frac{\partial}{\partial x_j}\tau_{ij} + \rho f_i\ . \tag{1.2.22}$$

For a compressible, Newtonian fluid, we have

$$\rho\big(\frac{\partial}{\partial t}\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u}\big) = -\nabla p + \mu\triangle\mathbf{u} + (\frac{\mu}{3}+\mu')\nabla\mathrm{div}(\mathbf{u}) + \rho\mathbf{f}\ . \tag{1.2.23}$$

1.2.3 Conservation of energy

Let us denote by E the total energy (per unit volume) of a moving volume V, i.e.

$$E = \rho(e_{\mathrm{in}} + \frac{|\mathbf{u}|^2}{2})\ , \tag{1.2.24}$$

where $\rho e_{\mathrm{in}} : D\times\bar{J}\to\mathbb{R}$ is the internal energy and $\rho\frac{|\mathbf{u}|^2}{2} : D\times\bar{J}\to\mathbb{R}_{\geq 0}$ is the kinetic energy.
The rate of change of total energy experienced by the volume is due to the mechanical work done by the surface and body forces and to heat fluxes and heat sources:

$$\begin{aligned}\frac{\mathcal{D}}{\mathcal{D}t}E &= \sum_i W_i + \sum_i \dot{Q}_i \\ &= \int_{\partial V}(\boldsymbol{\sigma}\mathbf{u})\cdot\mathbf{n}\,dS + \int_V \rho\mathbf{f}\cdot\mathbf{u}\,dV \\ &\quad - \int_{\partial V}\mathbf{q}\cdot\mathbf{n}\,dS + \int_V \rho q_V\,dV\ ,\end{aligned} \tag{1.2.25}$$

where $\mathbf{q} : D\times\bar{J}\to\mathbb{R}^d$ is the heat flux and $q_V : D\times\bar{J}\to\mathbb{R}$ is a heat source/sink [11].
We have: $[\rho e_{in}] = [\rho\frac{|\mathbf{u}|^2}{2}] = \frac{\mathrm{J}}{\mathrm{m}^3}$, $\mathbf{q} = \frac{\mathrm{J}}{\mathrm{m}^2\mathrm{s}}$, $q_V = \frac{\mathrm{J}}{\mathrm{kg\ s}}$.

With (1.1.10), we may rewrite this in the form of a conservation law:

for any control volume $V\subset D$:

$$\begin{aligned}&\underbrace{\int_V \frac{\partial}{\partial t}(\rho(e_{\mathrm{in}} + \frac{|\mathbf{u}|^2}{2}))\,dV}_{\text{energy change inside the volume}} + \underbrace{\int_{\partial V}\rho(e_{\mathrm{in}} + \frac{|\mathbf{u}|^2}{2})\mathbf{u}\cdot\mathbf{n}\,dS}_{\text{energy flux through the surface}} \\ = &\underbrace{\int_{\partial V}(\boldsymbol{\sigma}\mathbf{u})\cdot\mathbf{n}\,dS}_{\text{work from surface forces}} + \underbrace{\int_V\rho\mathbf{f}\cdot\mathbf{u}\,dV}_{\text{work from body forces}} - \underbrace{\int_{\partial V}\mathbf{q}\cdot\mathbf{n}\,dS}_{\text{heat fluxes}} + \underbrace{\int_V\rho q_V\,dV}_{\text{heat sources/sinks}}\ .\end{aligned} \tag{1.2.26}$$

This leads to the following partial differential equation, also called *energy* equation:

$$\begin{aligned}\frac{\partial}{\partial t}\big(\rho(e_{\mathrm{in}} + \frac{|\mathbf{u}|^2}{2})\big) + \mathrm{div}\big(\rho(e_{\mathrm{in}} + \frac{|\mathbf{u}|^2}{2})\mathbf{u}\big) \\ = \rho\mathbf{f}\cdot\mathbf{u} + \mathrm{div}(\boldsymbol{\sigma}\mathbf{u}) - \mathrm{div}(\mathbf{q}) + \rho q_V\ .\end{aligned} \tag{1.2.27}$$

Inserting the continuity equation (1.2.4) into the energy equation leads to

$$\begin{aligned}\rho\frac{\partial}{\partial t}e_{\text{in}} + \rho\mathbf{u}\cdot\nabla e_{\text{in}} + \rho\frac{\partial}{\partial t}(\frac{|\mathbf{u}|^2}{2}) + \rho\mathbf{u}\cdot\nabla(\frac{|\mathbf{u}|^2}{2}) \\ = \rho\mathbf{f}\cdot\mathbf{u} + \text{div}(\boldsymbol{\sigma}\mathbf{u}) - \text{div}(\mathbf{q}) + \rho q_V \ .\end{aligned} \tag{1.2.28}$$

We can rewrite (1.2.28) using Einstein's summation convention and get

$$\begin{aligned}\rho\frac{\partial}{\partial t}\big(e_{\text{in}} + \frac{u_j u_j}{2}\big) + \rho u_i\frac{\partial}{\partial x_i}\big(e_{\text{in}} + \frac{u_j u_j}{2}\big) \\ = \rho f_i u_i + \frac{\partial}{\partial x_i}(u_j\sigma_{ij}) - \frac{\partial q_i}{\partial x_i} + \rho q_V \ .\end{aligned} \tag{1.2.29}$$

By multiplying the momentum equation (1.2.21) with $\mathbf{u}$, we get a conservation equation for the kinetic energy:

$$\begin{aligned}\rho\frac{\mathcal{D}}{\mathcal{D}t}(\frac{u_j u_j}{2}) &= \rho\frac{\partial}{\partial t}\big(\frac{u_j u_j}{2}\big) + \rho u_i\frac{\partial}{\partial x_i}\big(\frac{u_j u_j}{2}\big) \\ &= \rho f_i u_i + u_j\frac{\partial\sigma_{ij}}{\partial x_i} \\ &= \rho f_i u_i - u_j\frac{\partial p}{\partial x_j} + u_j\frac{\partial\tau_{ij}}{\partial x_i} \\ &= \rho f_i u_i - u_j\frac{\partial p}{\partial x_j} + \frac{\partial}{\partial x_i}(u_j\tau_{ij}) - \tau_{ij}\frac{\partial u_j}{\partial x_i} \ .\end{aligned} \tag{1.2.30}$$

By subtracting (1.2.30) from (1.2.29), we get a conservation equation for the internal energy:

$$\begin{aligned}\rho\frac{\mathcal{D}e_{\text{in}}}{\mathcal{D}t} &= \rho\frac{\partial}{\partial t}\big(e_{\text{in}}\big) + \rho u_i\frac{\partial}{\partial x_i}\big(e_{\text{in}}\big) \\ &= \sigma_{ij}\frac{\partial u_j}{\partial x_i} - \frac{\partial q_i}{\partial x_i} + \rho q_V \\ &= -p\frac{\partial u_j}{\partial x_j} + \tau_{ij}\frac{\partial u_j}{\partial x_i} - \frac{\partial q_i}{\partial x_i} + \rho q_V \ .\end{aligned} \tag{1.2.31}$$

We remark that:

- The term $\tau_{ij}\frac{\partial u_j}{\partial x_i}$ appears in both equations (1.2.31) and (1.2.30), but with opposite signs. It stands for the dissipation of kinetic energy into heat [11].
- We can use Fourier's law to get the heat flux as a function of the temperature [11]:

$$\mathbf{q} = -\kappa\nabla T \ , \tag{1.2.32}$$

  where $T : D\times\bar{J}\to\mathbb{R}_{>0}$ is the temperature and $\kappa\in\mathbb{R}_{>0}$ (that we assume to be constant) is the fluid's heat conductivity. We have: $[\kappa] = \frac{\text{J}}{\text{Kms}}$.

Energy equation for incompressible flows

For incompressible flows, we have the divergence-free condition (1.2.6), such that the pressure work term $-p\frac{\partial u_j}{\partial x_j}$ inside equation (1.2.31) disappears. We do not need then to differentiate between isochoric (C_v) and isobaric (C_p) specific heat capacity [11], such that we can write the differential of e_{in} as

$$\mathcal{D}e_{\text{in}} = C\mathcal{D}T \ . \tag{1.2.33}$$

The equation for the internal energy becomes then an equation for the temperature T [11]:

$$\rho C\frac{\mathcal{D}T}{\mathcal{D}t} = \rho C\big(\frac{\partial T}{\partial t} + u_i\frac{\partial T}{\partial x_i}\big) = \kappa\triangle T + \tau_{ij}\frac{\partial u_j}{\partial x_i} + \rho q_V \ . \tag{1.2.34}$$

1.3 Navier-Stokes equations for an incompressible, homogeneous Newtonian fluid

For the remainder of this thesis, we assume that the fluid is incompressible and homogeneous, i.e. the density ρ is constant in space and time, which is a reasonable approximation for small Mach numbers, e.g. $\mathrm{Ma} < 0.3$ [11]. We further assume that the fluid is Newtonian with constant dynamic viscosity coefficient μ, and set

$$\nu = \frac{\mu}{\rho} , \tag{1.3.1}$$

where $\nu > 0$ is called the *kinematic* viscosity coefficient. Also, we identify the pressure p with the *scaled* pressure $\frac{p}{\rho}$.

1.3.1 Navier-Stokes equations and pressure equation

We obtain then, from the continuity equation (1.2.4) and the momentum equation (1.2.21), the Navier-Stokes equations for an incompressible, homogeneous fluid:

$$\frac{\partial}{\partial t}\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla p + \nu\triangle\mathbf{u} + \mathbf{f} \tag{1.3.2a}$$

$$\mathrm{div}(\mathbf{u}) = 0 , \tag{1.3.2b}$$

with a given initial velocity field $\mathbf{u}(0) = \mathbf{u}_0$, where $\mathbf{u}_0 : D \to \mathbb{R}^d$ is divergence-free [2, chapt. 1.1].

By taking the divergence of the momentum equation (1.3.2a), we end up with a Poisson equation for the pressure:

$$\triangle p = -\mathrm{div}((\mathbf{u}\cdot\nabla)\mathbf{u}) + \mathrm{div}(\mathbf{f}) \overset{(1.3.2b)}{=} -\sum_{i,j=1}^{d} \frac{\partial u_i}{\partial x_j}\frac{\partial u_j}{\partial x_i} + \mathrm{div}(\mathbf{f}) . \tag{1.3.3}$$

The terms $\mathrm{div}(\frac{\partial \mathbf{u}}{\partial t})$ and $\mathrm{div}(\triangle\mathbf{u})$ disappear because of the continuity equation (1.3.2b).

Thus, the pressure field is "fully determined at each instant of time by the velocity field", and at "any given point in space, it is determined by the velocity field everywhere" [2, chapt. 2.2]. This is a consequence of the incompressibility assumption. The sound speed "becomes infinite and velocity fluctuations everywhere are coupled instantaneously" [2, chapt. 2.2]. As a consequence, in the constant-density Navier-Stokes equations, the pressure is interpreted as a Lagrange multiplier that maintains the divergence-free condition for the velocity field rather than as a "purely thermodynamic variable related to density and temperature by an equation of state" [14, chapt. 2.5].

1.3.2 Boundary value problems

The Navier-Stokes equations (1.3.2) must be supplemented with initial and boundary conditions that depend on the physical problem under consideration [2, chapt. 2.2].

We consider in this thesis two types of boundary conditions: the *no-slip* boundary condition and the *space-periodic* boundary condition. They are discussed in detail throughout [2]. The no-slip boundary condition (flow past a rigid boundary) is "one of the few that correspond to a physically accessible boundary condition" [2, chapt. 2.2]. The space-periodic case is "not a physically achievable one, but it is relevant on the physical side as a model for some flows and is needed in the study of homogeneous turbulence" [2, chapt. 2.2].

There it is assumed that walls are far from the region being studied and thus that the wall effects are not influencing [2, chapt. 2.2]. We will consider two distinct situations in the space-periodic case: when the average flow (over the space domain) is zero, and when it is not necessarily zero.

No-slip boundary condition

At a solid wall, we have

$$\mathbf{u} = \mathbf{U}_{\text{wall}} \tag{1.3.4a}$$

$$\mathbf{u} \cdot \mathbf{n} = 0 \,, \tag{1.3.4b}$$

where $\mathbf{U}_{\text{wall}}$ is the velocity of the wall and $\mathbf{n}$ is the outward unit normal to the wall. In the Eulerian representation, we get directly for the first derivative in time

$$\frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{n} = \frac{\partial}{\partial t}(\mathbf{u} \cdot \mathbf{n}) = 0 \,. \tag{1.3.5}$$

Since boundary particles 'stick' to the wall, the velocity change experienced by those particles cannot have a non-zero component normal to the wall, such that in the Lagrangian representation, we have similarly

$$\frac{\mathcal{D}\mathbf{u}}{\mathcal{D}t} \cdot \mathbf{n} = 0 \,. \tag{1.3.6}$$

We get then also for the advective term

$$(\frac{\mathcal{D}\mathbf{u}}{\mathcal{D}t} - \frac{\partial \mathbf{u}}{\partial t}) \cdot \mathbf{n} = ((\mathbf{u} \cdot \nabla)\mathbf{u}) \cdot \mathbf{n} = 0 \,. \tag{1.3.7}$$

This yields the following Neumann boundary condition for the pressure

$$\nabla p \cdot \mathbf{n} = (\mathbf{f} + \nu \triangle \mathbf{u}) \cdot \mathbf{n} \,, \tag{1.3.8}$$

and the pressure solution of the Poisson equation (1.3.3) is defined up to an additive constant [2, chapter 2.2]. The consistency condition

$$\int_{\partial D} \nabla p \cdot \mathbf{n} \, dS = \int_D \triangle p \, d\mathbf{x} \,, \tag{1.3.9}$$

where $S = \partial D$ is the surface boundary of D, is satisfied, because we have

$$\int_D -\text{div}((\mathbf{u} \cdot \nabla)\mathbf{u}) \, d\mathbf{x} = -\int_{\partial D} \underbrace{((\mathbf{u} \cdot \nabla)\mathbf{u}) \cdot \mathbf{n}}_{=0} \, dS = 0 \tag{1.3.10}$$

and

$$\int_{\partial D} \nu \triangle \mathbf{u} \cdot \mathbf{n} \, dS = \nu \int_D \underbrace{\text{div}(\triangle \mathbf{u})}_{=0} \, d\mathbf{x} = 0 \,. \tag{1.3.11}$$

In this thesis, we assume that the shape and the volume of the domain D occupied by the fluid are independent of time, such that the boundary ∂D is at rest, and we have

$$\mathbf{u} = \mathbf{0} \quad \text{on } \partial D \,. \tag{1.3.12}$$

Space-periodic boundary condition

In the space-periodic case, we assume that the fluid fills the entire space $\mathbb{R}^d$, but with the condition that

$$\mathbf{u}, \mathbf{f} \text{ and } p \text{ are } L_1\text{-periodic in each spatial coordinate, with } L_1 > 0 \tag{1.3.13}$$

and denote the domain by $D = (0, L_1)^d$. Here also, the pressure solution of the Poisson equation (1.3.3) is defined up to an additive constant [2, chapter 2.2].

Space-periodic boundary condition with vanishing space average

Assuming that the average flow is zero for all time, i.e.

$$\frac{1}{|D|}\int_D \mathbf{u}(\mathbf{x},t)\,d\mathbf{x} = 0\,, \tag{1.3.14}$$

is sometimes "useful" and leads to a "simpler mathematical description" [2, chapter 2.2]. A sufficient condition is that the initial velocity field and the body forces have zero space-average. Indeed, because of the periodic boundary condition, if we integrate the momentum equation (1.3.2) over D, we are left with the relation

$$\frac{\partial}{\partial t}\big(\frac{1}{|D|}\int_D \mathbf{u}(\mathbf{x},t)\,d\mathbf{x}\big) = \frac{1}{|D|}\int_D \mathbf{f}(\mathbf{x},t)\,d\mathbf{x}\,. \tag{1.3.15}$$

That is, if

$$\frac{1}{|D|}\int_D \mathbf{u}_0(\mathbf{x})\,d\mathbf{x} = 0 \quad \text{and} \quad \frac{1}{|D|}\int_D \mathbf{f}(\mathbf{x},t)\,d\mathbf{x} = 0, \tag{1.3.16}$$

then

$$\frac{1}{|D|}\int_D \mathbf{u}(\mathbf{x},t)\,d\mathbf{x} = 0 \quad \text{at all times } t \geq 0\,. \tag{1.3.17}$$

Initial condition

The Navier-Stokes equations (1.3.2) are supplemented with an initial velocity field $\mathbf{u}_0 = \mathbf{u}(0)$ which, for consistency reasons, has to be divergence-free and satisfy the boundary conditions of the problem being considered [2, chapter 2.2].

1.3.3 Non-dimensional form

It is sometimes useful, for "both physical discussions and mathematical transparency" [2, chapter 1.1], to consider the Navier-Stokes equations in their non-dimensional form.
For the sake of this section only, the pressure p is the 'real' pressure, and not the pressure scaled by the density.

Reynolds number similarity

Let L_* be a reference length and U_* a reference velocity of the flow. Typically, L_* characterizes the size of the domain and U_* characterizes the magnitude of the initial and boundary conditions for the velocity [14, chapt. 2.9]. Let us set

$$\mathbf{x}' = \frac{1}{L_*}\mathbf{x}, \quad t' = \frac{U_*}{L_*}t, \quad \mathbf{u}' = \frac{1}{U_*}\mathbf{u}, \quad p' = \frac{1}{\rho U_*^2}p, \quad \mathbf{f}' = \frac{L_*}{U_*^2}\mathbf{f}\,. \tag{1.3.18}$$

Then we obtain the non-dimensional form [2, chapt. 1.1] of the Navier-Stokes equations:

$$\frac{\partial \mathbf{u}'}{\partial t'} + (\mathbf{u}'\cdot\nabla')\mathbf{u}' = -\nabla' p' + \frac{1}{\mathrm{Re}}\triangle'\mathbf{u}' + \mathbf{f}' \tag{1.3.19a}$$

$$\nabla'\cdot\mathbf{u}' = 0\,, \tag{1.3.19b}$$

where

$$\mathrm{Re} = \frac{U_* L_*}{\nu} = \rho\frac{U_* L_*}{\mu} \tag{1.3.20}$$

is a non-dimensional number called the *Reynolds number*.

Thus, different experiments sharing the same Reynolds number yield similar results, i.e. they yield the same results up to rescaling. This is the so-called *Reynolds number similarity*,

which is "constantly used in mechanical engineering" [2, chapt. 1.1].

The Reynolds number can be seen as a measure of the ratio of inertial forces (of the order of $\frac{U_*^2}{L_*}$) over viscous forces (of the order of $\nu\frac{U_*}{L_*^2}$) on the largest scales of the flow [2, chapt. 1.1].

Space, time, rotational, reflectional and Galilean invariance

Suppose an experiment $\mathcal{E}$ is done on a coordinate system that is orientated differently than the reference experiment $\mathcal{E}'$ (by rotation or by reflection of a coordinate axis described by the orthogonal matrix $\mathbf{R}$), performed at time T later than the reference experiment, translated by an amount $\mathbf{X}_0$ from the reference experiment, and that is moving with constant velocity $\mathbf{V}_0$. With the following choice of variables

$$\begin{aligned} \mathbf{x}' &= \frac{1}{L_*}\mathbf{R}^T[\mathbf{x} - (\mathbf{X}_0 + \mathbf{V}_0(t-T))], \quad t' = \frac{U_*}{L_*}(t-T), \\ \mathbf{u}' &= \frac{1}{U_*}\mathbf{R}^T[\mathbf{u} - \mathbf{V}_0], \quad p' = \frac{1}{\rho U_*^2}p, \quad \mathbf{f}' = \frac{L_*}{U_*^2}\mathbf{R}^T\mathbf{f}\ , \end{aligned} \tag{1.3.21}$$

we obtain again the non-dimensional equations (1.3.19). Thus, "just like all phenomena described by classical mechanics, the behavior of fluid flows is the same in all inertial frames" [14, chapt. 2.9]. It is to note that it is the fluid's acceleration (and it's associated forces) that are Galilean invariant, and not the fluid's velocity. Also, although the velocity is invariant under rotation or reflection, the vorticity is in general not [14, chapt. 2.9].

If the coordinate system is moving with a variable velocity and/or is rotating, a fictitious force has generally to be added to the non-dimensional equations (1.3.19) in order to take into account the effect of the non-inertial coordinate system motion [14, chapt. 2.9]. These fictitious forces occur for example when an object is accelerated in a fluid 'at rest' [11], or in "meteorology and turbomachinery" [14, chapt. 2.9].

1.4 Elements of the mathematical theory of the Navier-Stokes equations

In this section, we follow the presentation given in [2], and, except if specified otherwise, the statements refer to results as they are found in [2].

First, we introduce function spaces that are "appropriate for use in mathematical treatments of the Navier-Stokes equations", but that are also physically meaningful [2, chapt. 2.5].

1.4.1 Kinetic energy and enstrophy, function spaces

The kinetic energy (divided by the density ρ) of a fluid with velocity field $\mathbf{u}$ and occupying a region D is given by

$$e(\mathbf{u}) = \frac{1}{2}\int_D |\mathbf{u}|^2\, d\mathbf{x}\ . \tag{1.4.1}$$

Another important integral quantity is the enstrophy

$$E(\mathbf{u}) = \sum_{i=1}^{d}\int_D |\nabla u_i|^2\, d\mathbf{x} = \sum_{i,j=1}^{d}\int_D \left|\frac{\partial u_i}{\partial x_j}\right|^2 d\mathbf{x}\ , \tag{1.4.2}$$

as we see next.

For the boundary conditions we consider (no-slip and space-periodic boundary conditions), the equation for the conservation of kinetic energy becomes

$$\frac{\mathrm{d}}{\mathrm{d}t}e(\mathbf{u}) + \nu E(\mathbf{u}) = \int_D \mathbf{f}\cdot\mathbf{u}\,d\mathbf{x}\,. \tag{1.4.3}$$

Indeed, by integrating the different terms of the equation for the kinetic energy (1.2.30), we get, using Einstein's summation convention,

$$\begin{aligned}
\int_D \frac{\partial}{\partial t}(\frac{u_j u_j}{2})\,d\mathbf{x} &= \frac{\partial}{\partial t}\int_D \frac{\partial}{\partial t}(\frac{u_j u_j}{2})\,d\mathbf{x} = \frac{\mathrm{d}}{\mathrm{d}t}e(\mathbf{u}),\\
\int_D u_i\frac{\partial}{\partial x_i}(\frac{u_j u_j}{2})\,d\mathbf{x} &= -\int_D \underbrace{\frac{\partial u_i}{\partial x_i}}_{=0}\frac{u_j u_j}{2}\,d\mathbf{x} + \underbrace{\int_{\partial D}\mathbf{u}\cdot\mathbf{n}\frac{u_j u_j}{2}\,dS}_{=0 \text{ for per. and no-slip b.c.}} = 0,\\
\int_D -u_j\frac{\partial p}{\partial x_j}\,d\mathbf{x} &= \int_D \underbrace{\frac{\partial u_i}{\partial x_i}}_{=0} p\,d\mathbf{x} - \underbrace{\int_{\partial D}\mathbf{u}\cdot\mathbf{n}p\,dS}_{=0 \text{ for per. and no-slip b.c.}} = 0,\\
\int_D u_i\nu\frac{\partial^2 u_i}{\partial x_j^2}\,d\mathbf{x} &= -\nu\int_D (\frac{\partial u_i}{\partial x_j})^2\,d\mathbf{x} + \underbrace{\int_{\partial D}(\nabla u_i\cdot\mathbf{n})u_i\,dS}_{=0 \text{ for per. and no-slip b.c.}} = -\nu E(\mathbf{u})\,.
\end{aligned} \tag{1.4.4}$$

We recall that the pressure p is here scaled by the density.

Also, for the boundary conditions we consider, the enstrophy can be written as

$$\begin{aligned}
E(\mathbf{u}) &= \int_D \sum_{i,j=1}^{d}\left|\frac{\partial u_i}{\partial x_j}\right|^2 d\mathbf{x}\\
&= \int_D |\boldsymbol{\omega}|^2\,d\mathbf{x} + \int_D \sum_{i=1}^{d}(\partial_{x_i}\mathbf{u})\cdot\nabla u_i\,d\mathbf{x}\\
&= \int_D |\boldsymbol{\omega}|^2\,d\mathbf{x} - \int_D \sum_{i=1}^{d}\mathrm{div}(\partial_{x_i}\mathbf{u})u_i\,d\mathbf{x} + \int_{\partial D}\sum_{i=1}^{d}u_i(\partial_{x_i}\mathbf{u})\cdot\mathbf{n}\,dS\\
&= \int_D |\boldsymbol{\omega}|^2\,d\mathbf{x} - \int_D \sum_{i=1}^{d}\partial_{x_i}(\underbrace{\mathrm{div}(\mathbf{u})}_{=0})u_i\,d\mathbf{x} + \underbrace{\int_{\partial D}((\mathbf{u}\cdot\nabla)\mathbf{u})\cdot\mathbf{n}\,dS}_{=0 \text{ for per. and no-slip b.c.}}\\
&= \int_D |\boldsymbol{\omega}|^2\,d\mathbf{x}\,,
\end{aligned} \tag{1.4.5}$$

where

$$\boldsymbol{\omega} = \mathrm{rot}(\mathbf{u}) = \nabla\times\mathbf{u} = \begin{pmatrix}\frac{\partial u_3}{\partial x_2} - \frac{\partial u_2}{\partial x_3}\\ \frac{\partial u_1}{\partial x_3} - \frac{\partial u_3}{\partial x_1}\\ \frac{\partial u_2}{\partial x_1} - \frac{\partial u_1}{\partial x_2}\end{pmatrix} \tag{1.4.6}$$

is the vorticity vector (which can be heuristically interpreted as twice the averaged, at a given instant in time, angular velocity of a fluid particle [11]; in two dimensions, it has only one non-zero component). Thus, for these boundary conditions, the enstrophy is the integral of the square of the vorticity over the domain.

When there are no volume forces (i.e. when $\mathbf{f} = 0$), the kinetic energy decays "by viscous effect" [2, chapt. 2.1] at the rate $-\nu E(\mathbf{u})$:

$$\frac{\mathrm{d}}{\mathrm{d}t}e(\mathbf{u}) = -\nu E(\mathbf{u}) \,. \tag{1.4.7}$$

Function spaces

As remarked in [2, chapt. 2.1], physical solutions of the Navier-Stokes equations should have finite kinetic energy and finite enstrophy. We consider thus the two spaces H and V that take into account the boundary conditions, the incompressibility assumption and the boundedness of the physical quantities $e(\mathbf{u})$ and $E(\mathbf{u})$. The space H is the space of incompressible vector fields with finite kinetic energy and with appropriate boundary conditions, and V is the space of incompressible vector fields with finite enstrophy and also with appropriate boundary conditions.

We assume here the domain $D \subset \mathbb{R}^d$, for $d = 2, 3$, to be open, bounded and connected, and its boundary ∂D is assumed to be either C^2 or D is assumed to be convex, in order to ensure local $H^2(D)$ regularity of the velocity field ([1, sect. 3.1] and references there).

Consider the space $L^2(D)$ of square integrable vector fields from D into $\mathbb{R}^d$, which is a Hilbert space with the inner product

$$(\mathbf{u}, \mathbf{v}) = \int_D \mathbf{u} \cdot \mathbf{v} \, d\mathbf{x} \tag{1.4.8}$$

and the associated norm

$$|\mathbf{u}| = (\mathbf{u}, \mathbf{u})^{\frac{1}{2}} = \big(\int_D |\mathbf{u}|^2 \, d\mathbf{x} \big)^{\frac{1}{2}} \,. \tag{1.4.9}$$

We have the relation $|\mathbf{u}|^2 = 2e(\mathbf{u})$, such that $L^2(D)$ consists of the space of all velocity fields with finite kinetic energy.

Further, we consider the Sobolev space $H^1(D)$ of vector fields on D that are square integrable and whose gradient is also square integrable. This is a Hilbert space with the inner product

$$((\mathbf{u}, \mathbf{v}))_1 = \frac{1}{L_1^2} \underbrace{\int_D \mathbf{u} \cdot \mathbf{v} \; d\mathbf{x}}_{=(\mathbf{u},\mathbf{v})} + \underbrace{\int_D \sum_{i=1}^{d} \frac{\partial \mathbf{u}}{\partial x_i} \cdot \frac{\partial \mathbf{v}}{\partial x_i} \, d\mathbf{x}}_{:=((\mathbf{u},\mathbf{v}))} \,, \tag{1.4.10}$$

where L_1 is a typical length, e.g. the diameter of D ($L_1 = 1$ for non-dimensional variables) [2, chapt. 1.4], and the associated norm is given by

$$||\mathbf{u}||_1 = ((\mathbf{u}, \mathbf{u}))_1^{\frac{1}{2}} = \big(\frac{1}{L_1^2} \underbrace{\int_D |\mathbf{u}|^2 \, d\mathbf{x}}_{|\mathbf{u}|^2} + \underbrace{\int_D \sum_{i,j=1}^{d} \left| \frac{\partial u_i}{\partial x_j} \right|^2 \, d\mathbf{x}}_{:=||\mathbf{u}||^2} \big)^{\frac{1}{2}} \,. \tag{1.4.11}$$

We have the relation $||\mathbf{u}||^2 = E(\mathbf{u})$, such that $H^1(D)$ consists of the space of all velocity fields with finite enstrophy. From (1.4.11), the following inequality holds:

$$|\mathbf{u}|^2 \leq L_1^2 ||\mathbf{u}||_1^2 \,. \tag{1.4.12}$$

Function spaces for the no-slip boundary conditions

In the no-slip case, we have

$$H = H_{dir} = \{\mathbf{u} \in L^2(D) : \nabla \cdot \mathbf{u} = 0, \mathbf{u} \cdot \mathbf{n}|_{\partial D} = 0\} \tag{1.4.13}$$

and

$$V = V_{dir} = \{\mathbf{u} \in H^1(D) : \nabla \cdot \mathbf{u} = 0, \mathbf{u}|_{\partial D} = 0\} \, . \tag{1.4.14}$$

Here, $\mathbf{n}$ denotes the outward unit normal to the domain D which is defined almost everywhere on the Lipschitz boundary ∂D [1, sect. 3.1]. Since D is bounded and $\mathbf{u} \in V_{dir}$ vanishes at the boundary, we can use the Poincaré inequality

$$|\mathbf{u}|^2 \leq \frac{1}{\lambda_1} ||\mathbf{u}||^2 \quad \text{for all } \mathbf{u} \in V_{dir} \, , \tag{1.4.15}$$

where $\lambda_1 > 0$ is the smallest eigenvalue of the corresponding Stokes operator [2, chapt. 2.5] (see Section 1.4.4). Then for $\mathbf{u} \in V_{dir}$, we have

$$||\mathbf{u}||_1^2 = \frac{1}{L_1^2} |\mathbf{u}|^2 + ||\mathbf{u}||^2 \leq (\frac{1}{L_1^2} \frac{1}{\lambda_1} + 1) ||\mathbf{u}||^2 \, , \tag{1.4.16}$$

such that the semi-norm ('enstrophy norm') $|| \cdot ||$ associated to the inner product $((\cdot, \cdot))$ is in this case actually a norm.

We endow then H_{dir} with the norm $| \cdot |_H = | \cdot |$ and the associated inner product $(\cdot, \cdot)_H = (\cdot, \cdot)$, and we endow V_{dir} with the norm $| \cdot |_V = || \cdot ||$ and the associated inner product $(\cdot, \cdot)_V = ((\cdot, \cdot))$.

Function spaces for the periodic boundary conditions with vanishing space average

In the periodic case with vanishing space average, we have

$$H = \dot{H}_{per} = \{\mathbf{u} \in L^2_{per}(D) : \nabla \cdot \mathbf{u} = 0, \int_D \mathbf{u} \, d\mathbf{x} = 0\} \tag{1.4.17}$$

and

$$V = \dot{V}_{per} = \{\mathbf{u} \in H^1_{per}(D) : \nabla \cdot \mathbf{u} = 0, \int_D \mathbf{u} \, d\mathbf{x} = 0\} \, . \tag{1.4.18}$$

The Poincaré inequality can also be used in this case, because $\mathbf{u} \in V_{per}$ has a zero space average, and we have

$$|\mathbf{u}|^2 \leq \frac{1}{\lambda_1} ||\mathbf{u}||^2 \quad \text{for all } \mathbf{u} \in \dot{V}_{per} \, , \tag{1.4.19}$$

where $\lambda_1 > 0$ is the smallest eigenvalue of the corresponding Stokes operator [2, chapt. 2.5] (see Section 1.4.4). Then, similarly to the no-slip case, the semi-norm ('enstrophy norm') $|| \cdot ||$ associated to the inner product $((\cdot, \cdot))$ is actually a norm.

We endow $\dot{H}_{per}$ with the norm $| \cdot |_H = | \cdot |$ and the associated inner product $(\cdot, \cdot)_H = (\cdot, \cdot)$, and we endow $\dot{V}_{per}$ with the norm $| \cdot |_V = || \cdot ||$ and the associated inner product $(\cdot, \cdot)_V = ((\cdot, \cdot))$.

Function spaces for the periodic boundary conditions

In the general periodic case, we have

$$H = H_{per} = \{\mathbf{u} \in L^2_{per}(D) : \nabla \cdot \mathbf{u} = 0\} \tag{1.4.20}$$

and

$$V = V_{per} = \{\mathbf{u} \in H^1_{per}(D) : \nabla \cdot \mathbf{u} = 0\} \ . \tag{1.4.21}$$

The Poincaré inequality is not valid in this case [2, chapt. 2.5], because $\mathbf{u} \in V_{per}$ does not necessarily vanish at the boundary and does not have a zero space average. Hence we endow V_{per} with the full norm $|\cdot|_V = ||\cdot||_1$ and associated inner product $((\cdot,\cdot))_1$, and it holds the inequality (1.4.12) instead. H_{per} is endowed with the norm $|\cdot|_H = |\cdot|$ and the associated inner product $(\cdot,\cdot)_H$.

Fourier Series

The spaces H_{per}, V_{per}, $\dot{H}_{per}$ and $\dot{V}_{per}$ can be characterized in terms of Fourier series [2, chapt. 2.5]. We have

$$H_{per} = \{\mathbf{u} = \sum_{\mathbf{k}\in\mathbb{Z}^d} \hat{\mathbf{u}}_\mathbf{k} e^{i\frac{2\pi}{L_1}\mathbf{k}\cdot\mathbf{x}} : \underbrace{\hat{\mathbf{u}}_{-\mathbf{k}} = \bar{\hat{\mathbf{u}}}_\mathbf{k}}_{\Leftrightarrow \mathbf{u}\in\mathbb{R}^d}, \underbrace{\hat{\mathbf{u}}_\mathbf{k}\cdot\mathbf{k} = 0}_{\Leftrightarrow \nabla\cdot\mathbf{u}=0}, \underbrace{\sum_{\mathbf{k}\in\mathbb{Z}^d} |\hat{\mathbf{u}}_\mathbf{k}|^2 < \infty}_{=\frac{1}{|D|}|\mathbf{u}|}\} \ , \tag{1.4.22}$$

$$V_{per} = \{\mathbf{u} = \sum_{\mathbf{k}\in\mathbb{Z}^d} \hat{\mathbf{u}}_\mathbf{k} e^{i\frac{2\pi}{L_1}\mathbf{k}\cdot\mathbf{x}} : \hat{\mathbf{u}}_{-\mathbf{k}} = \bar{\hat{\mathbf{u}}}_\mathbf{k}, \hat{\mathbf{u}}_\mathbf{k}\cdot\mathbf{k} = 0, \underbrace{\sum_{\mathbf{k}\in\mathbb{Z}^d} (\frac{1}{L_1^2} + 2\pi|\mathbf{k}|^2)|\hat{\mathbf{u}}_\mathbf{k}|^2 < \infty}_{=\frac{1}{|D|}||\mathbf{u}||_1}\} \ , \tag{1.4.23}$$

$$\dot{H}_{per} = \{\mathbf{u} = \sum_{\mathbf{k}\in\mathbb{Z}^d\setminus\{0\}} \hat{\mathbf{u}}_\mathbf{k} e^{i\frac{2\pi}{L_1}\mathbf{k}\cdot\mathbf{x}} : \hat{\mathbf{u}}_{-\mathbf{k}} = \bar{\hat{\mathbf{u}}}_\mathbf{k}, \hat{\mathbf{u}}_\mathbf{k}\cdot\mathbf{k} = 0, \underbrace{\sum_{\mathbf{k}\in\mathbb{Z}^d\setminus\{0\}} |\hat{\mathbf{u}}_\mathbf{k}|^2 < \infty}_{=\frac{1}{|D|}|\mathbf{u}|}\} \ , \tag{1.4.24}$$

$$\dot{V}_{per} = \{\mathbf{u} = \sum_{\mathbf{k}\in\mathbb{Z}^d\setminus\{0\}} \hat{\mathbf{u}}_\mathbf{k} e^{i\frac{2\pi}{L_1}\mathbf{k}\cdot\mathbf{x}} : \hat{\mathbf{u}}_{-\mathbf{k}} = \bar{\hat{\mathbf{u}}}_\mathbf{k}, \hat{\mathbf{u}}_\mathbf{k}\cdot\mathbf{k} = 0, \underbrace{\sum_{\mathbf{k}\in\mathbb{Z}^d\setminus\{0\}} |\mathbf{k}|^2|\hat{\mathbf{u}}_\mathbf{k}|^2 < \infty}_{=\frac{1}{2\pi|D|}||\mathbf{u}||}\} \ . \tag{1.4.25}$$

In the following, we shall use the symbols V and H in all statements which apply generically, i.e. to either choice of V and of H. In all cases, we have the dense inclusions $V \subset H$ [1, sect. 3.1] and the corresponding norms are related by the inequality

$$|\mathbf{u}|_H^2 \leq \frac{1}{C_{HV}}|\mathbf{u}|_V^2 \quad \forall \mathbf{u} \in V \ , \tag{1.4.26}$$

where for the no-slip case and the space-periodic case with vanishing space-average, $C_{HV} = \lambda_1$, which is the smallest eigenvalue of the corresponding Stokes operator (see Section 1.4.4), and for the general space-periodic case, $C_{HV} = \frac{1}{L_1^2}$.

We present next the Helmholtz-Leray decomposition of vector fields, which allows us to write the Navier-Stokes equations in functional form and to define the Stokes operator.

1.4.2 Helmholtz-Leray decomposition of vector fields

The Helmholtz-Leray decomposition resolves a vector field $\mathbf{w} \in L^2(D)$ on a bounded set $D \subset \mathbb{R}^d$ into the sum of a gradient and a curl vector, by taking into account the boundary conditions of the problem. It is a generalization of the Helmholtz decomposition, which is done on the whole space $\mathbb{R}^d$ without any boundary conditions [2, chapt. 2.3].

The decomposition is of the form

$$\mathbf{w} = \nabla q + \mathbf{v}, \quad \text{with } \operatorname{div}(\mathbf{v}) = 0 \,, \tag{1.4.27}$$

which implies that

$$\triangle q = \operatorname{div}(\mathbf{w}) \tag{1.4.28}$$

and that, at least locally, $\mathbf{v}$ is a curl vector [2, chapt. 2.3].

One can calculate the decomposition by solving (1.4.28) to get q (with boundary conditions on q that depend on those of $\mathbf{w}$), and then use the relation

$$\mathbf{v} = \mathbf{w} - \nabla q \tag{1.4.29}$$

to get $\mathbf{v}$.

Space-periodic boundary conditions

In the space-periodic case, $\mathbf{w}$ is periodic, so we require $\mathbf{v}$ to be periodic as well and impose periodic boundary conditions on q, which together with Equation (1.4.28) determine q uniquely in terms of $\mathbf{w}$ (up to an additive constant) [2, chapt. 2.3].

No-slip boundary conditions

In the no-slip case, we only require that

$$\mathbf{v} \cdot \mathbf{n} = 0 \text{ on } \partial D \,, \tag{1.4.30}$$

which implies that

$$\nabla q \cdot \mathbf{n} = \mathbf{w} \cdot \mathbf{n} \text{ on } \partial D \,. \tag{1.4.31}$$

Together with Equation (1.4.28), this determines q uniquely in terms of $\mathbf{w}$ (up to an additive constant) [2, chapt. 2.3].

It is worth to note that, "contrary to the usual Helmholtz decomposition", the Helmholtz-Leray decomposition is unique (up to an additive constant for q) [2, chapt. 2.3], and the map

$$P_L : \begin{cases} L^2(D) & \to H \\ \mathbf{w} & \mapsto \mathbf{v}(\mathbf{w}) \end{cases} \tag{1.4.32}$$

is well-defined. This map, so-called *Leray projector*, is an orthogonal projector from $L^2(D)$ onto H [2, chapt. 5.0]. In particular, if $\mathbf{w}$ is already divergence-free and satisfies the boundary conditions characterizing H, then $P_L\mathbf{w} = \mathbf{w}$. And if $\mathbf{w}$ is a gradient vector which is not divergence-free, then $P_L\mathbf{w} = \mathbf{0}$.

1.4.3 Functional evolution equation for the velocity field

By applying the Leray projector on the momentum equation (1.3.2), one finds [2, chapt. 2.3] that

$$\frac{\partial \mathbf{u}}{\partial t} + \nu A\mathbf{u} + B(\mathbf{u}) = P_L\mathbf{f}, \quad \text{and } P_L\nabla p = 0 \,, \tag{1.4.33}$$

where

$$A\mathbf{u} = -P_L\triangle\mathbf{u}, \quad B(\mathbf{u}) = B(\mathbf{u},\mathbf{u}), \quad B(\mathbf{u},\mathbf{v}) = P_L((\mathbf{u}\cdot\nabla)\mathbf{v}) \,. \tag{1.4.34}$$

The operator A is the *Stokes operator* [2, chapt. 2.3]. In the space-periodic case, we have

$$A\mathbf{u} = -P_L\triangle\mathbf{u} = -\triangle\mathbf{u} \,. \tag{1.4.35}$$

However, in the no-slip case, it holds

$$A\mathbf{u} = -P_L\triangle\mathbf{u} \neq -\triangle\mathbf{u} \tag{1.4.36}$$

in general [2, chapt. 2.3].

We assume further that $\mathbf{f}$ belongs to H. If not, we set $\mathbf{f} = P_L\mathbf{f}$ and add the term $(I-P_L)\mathbf{f}$ (which is a gradient vector) to the pressure gradient, which disappears in (1.4.33). Then we can write the nonlinear dynamical system

$$\mathbf{u}'(t) = \mathbf{F}(t,\mathbf{u}(t)) \,, \tag{1.4.37}$$

where

$$\mathbf{u}' = \frac{\partial \mathbf{u}}{\partial t}, \quad \mathbf{F}(t,\mathbf{u}) = \mathbf{f}(t) - \nu A\mathbf{u} - B(\mathbf{u}) \,. \tag{1.4.38}$$

Functional formulation of the Navier-Stokes equations

The system (1.4.37) yields the following *functional formulation* [2, chapt. 5.0] of the Navier-Stokes equations:

$$\begin{gathered} \text{given } T>0, \mathbf{u}_0 \in H \text{ and } \mathbf{f} \in L^2(J;H), \\ \text{find } \mathbf{u} \in L^\infty(J;H)\cap L^2(J;V) \text{ with } \mathbf{u}' \in L^1(J;V^*), \text{ such that} \\ \mathbf{u}' = \mathbf{f} - \nu A\mathbf{u} - B(\mathbf{u}) \,, \end{gathered} \tag{1.4.39}$$

where V^* is the dual of V.

1.4.4 The Stokes operator

The Stokes operator is associated with the linear part of the Navier-Stokes equations, and as such, plays an "important role in the study of the full, nonlinear equations" [2, chapt. 2.6]. We have

$$A\mathbf{u} = -P_L\triangle\mathbf{u} \quad \text{for } \mathbf{u} \in D(A) = V\cap H^2(D) \,, \tag{1.4.40}$$

where $D(A)$ is the domain of A, i.e. the subspace of H for which $A\mathbf{u}$ is meaningful.

In the no-slip case and in the space-periodic case with vanishing space average, it has been shown ([2, chapt. 2.6] and references there) that

$$(A\mathbf{u},\mathbf{v})_H = (\mathbf{u},\mathbf{v})_V \quad \text{for all } \mathbf{u},\mathbf{v} \in D(A) \,, \tag{1.4.41}$$

and thus that the Stokes operator A is self-adjoint, i.e.

$$(A\mathbf{u},\mathbf{v})_H = (\mathbf{u},A\mathbf{v})_H \quad \text{for all } \mathbf{u},\mathbf{v} \in D(A) \,, \tag{1.4.42}$$

and positive definite, i.e.

$$(A\mathbf{u},\mathbf{u})_H = (\mathbf{u},\mathbf{u})_V = |\mathbf{u}|_V^2 > 0 \quad \text{for all } \mathbf{u} \neq 0 \text{ in } D(A)\ . \tag{1.4.43}$$

More precisely, the Stokes operator is a closed, unbounded, self-adjoint positive definite operator on its domain $D(A)$ [1, sect. 7.1]. By the spectral theorem, A has a discrete spectrum $\Sigma = (\lambda_m, m \in \mathbb{N}) \subset \mathbb{R}_{>0}$ which consists of real eigenvalues

$$0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m \leq \cdots\ , \quad \lambda_m \to +\infty \text{ as } m \to +\infty\ , \tag{1.4.44}$$

which accumulate only at infinity, and which admits a countable sequence of eigenfunctions $(\mathbf{w}_m, m \in \mathbb{N})$, with

$$A\mathbf{w}_m = \lambda_m \mathbf{w}_m, \quad m = 1, 2, ...\quad , \tag{1.4.45}$$

which are dense in H and V and constitute an orthonormal basis of H [1, sect. 7.1].

The first eigenvalue λ_1 is exactly the best constant [2, chapt. 2.6] for the Poincaré inequality

$$|\mathbf{u}|_H^2 \leq \frac{1}{\lambda_1}|\mathbf{u}|_V^2\ , \tag{1.4.46}$$

that we introduced in (1.4.15) and (1.4.19). The asymptotic behavior of the eigenvalues is given [2, chapt. 2.6] by

$$\lambda_m \sim \lambda_1 m^{\frac{2}{d}} \quad \text{as} \quad m \to \infty\ . \tag{1.4.47}$$

Because $(\mathbf{w}_m, m \in \mathbb{N})$ is an orthonormal basis in H, we may write for $\mathbf{u} \in H$

$$\mathbf{u} = \sum_{m=1}^{\infty} \hat{u}_m \mathbf{w}_m, \quad \hat{u}_m = (\mathbf{u}, \mathbf{w}_m)_H\ . \tag{1.4.48}$$

We have, from the orthonormality property and (1.4.43),

$$|\mathbf{u}|_H^2 = \sum_{m=1}^{\infty} |\hat{u}_m|^2 \tag{1.4.49}$$

and

$$|\mathbf{u}|_V^2 = \sum_{m=1}^{\infty} \lambda_m |\hat{u}_m|^2\ . \tag{1.4.50}$$

Since A is a positive, self-adjoint operator, we can define fractional powers of A [1, sect. 3.2]. We denote the fractional powers by A^a, for $a \in \mathbb{R}$, and by $D(A^a)$ the domain of A^a. The powers A^a are defined by

$$A^a \mathbf{u} = \sum_{m=1}^{\infty} \lambda_m^a \hat{u}_m \mathbf{w}_m \tag{1.4.51}$$

and

$$\mathbf{u} \in D(A^a) \Leftrightarrow |\mathbf{u}|_{D(A^a)} = \sum_{m=1}^{\infty} \lambda_m^{2a} |\hat{u}_m|^2 < \infty\ . \tag{1.4.52}$$

We have then $D(A^{\frac{1}{2}}) = V$. Furthermore, it holds that $V^* = D(A^{-\frac{1}{2}})$, where V^* is the dual of V [2, chapt. 2.6].

Expression for the eigenfunctions in the periodic case with vanishing space average

In the periodic case with vanishing space average, the eigenfunctions $(\mathbf{w}_m, m \in \mathbb{N})$ can be expressed from their Fourier expansion (1.4.24), i.e.

$$\mathbf{w}_\mathbf{k} = \mathbf{a}_\mathbf{k} e^{i\frac{2\pi}{L_1}\mathbf{k}\cdot\mathbf{x}} + \bar{\mathbf{a}}_\mathbf{k} e^{-i\frac{2\pi}{L_1}\mathbf{k}\cdot\mathbf{x}} , \tag{1.4.53}$$

where for each $\mathbf{k}$ the $\mathbf{a}_\mathbf{k}$ are $d-1$ independent vectors in $\mathbb{C}^d$ such that $\mathbf{a}_\mathbf{k}\cdot\mathbf{k} = 0$ and with $\mathbf{a}_{-\mathbf{k}} = \bar{\mathbf{a}}_\mathbf{k}$. The eigenvalues [2, chapt. 2.6] are

$$\lambda_\mathbf{k} = \frac{4\pi^2}{L_1^2}|\mathbf{k}|^2 . \tag{1.4.54}$$

They can be ordered in nondecreasing order such that, for each $\lambda_\mathbf{k}$, $\mathbf{k} \in \mathbb{Z}^d\backslash\{0\}$, we have a corresponding eigenvalue λ_m for some $m \in \mathbb{N}$, with $\lambda_m \leq \lambda_{m+1}$. The corresponding eigenfunction is $\mathbf{w}_m = \mathbf{w}_\mathbf{k}$. The eigenfunctions have been explicitly calculated in [16, append. A.1] and are presented in Section 1.5.

In the space-periodic case without vanishing space-average, the Stokes operator A is not positive definite anymore [2, chapt. 2.6]. However, we can consider instead the operator $\tilde{A}$ defined by

$$\tilde{A}\mathbf{u} = \frac{1}{L_1^2}\mathbf{u} + A\mathbf{u} \quad \text{for } \mathbf{u} \in D(\tilde{A}) \equiv D(A) . \tag{1.4.55}$$

One can show [2, chapt. 2.6] that

$$\begin{aligned}(\tilde{A}\mathbf{u},\mathbf{v})_H &= \frac{1}{L_1^2}(\mathbf{u},\mathbf{v}) + ((\mathbf{u},\mathbf{v})) \\ &= ((\mathbf{u},\mathbf{v}))_1 = (\mathbf{u},\mathbf{v})_V \quad \text{for all } \mathbf{u} \in D(A), \mathbf{v} \in V .\end{aligned} \tag{1.4.56}$$

It holds that $D(\tilde{A}^{\frac{1}{2}}) = V$ and that $D(\tilde{A}^{-\frac{1}{2}}) = V^*$, where V^* is the dual of V. Furthermore, $\tilde{A}$ is a positive self-adjoint operator with compact inverse, and possesses a sequence of positive eigenvalues $(\tilde{\lambda}_m, m \in \mathbb{N})$ associated with an orthonormal basis $(\mathbf{w}_m, m \in \mathbb{N})$ of H [2, chapt. 2.6]. We can recover the eigenvalues $(\lambda_m, m \in \mathbb{N})$ of the Stokes operator A, which are related to those of $\tilde{A}$, by

$$\lambda_m = \tilde{\lambda}_m - \frac{1}{L_1^2} . \tag{1.4.57}$$

We have

$$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m \leq \cdots , \quad \lambda_m \to +\infty \text{ as } m \to +\infty . \tag{1.4.58}$$

The eigenvalues and eigenfunctions are actually the same as in the case of the vanishing space average, except that now we include the case with the wavenumber vector $\mathbf{k} = \mathbf{0}$, which is associated with the eigenvalue $\lambda_1 = 0$ and with a d-dimensional eigenspace [2, chapt. 2.6].

1.4.5 Weak formulation of the Navier-Stokes equations

By multiplying the Navier-Stokes equations (1.3.2) with a test function and then integrating by parts, we obtain the *weak formulation* of the Navier-Stokes equations, which, equipped with either no-slip or space-periodic boundary conditions, is [2, chapt. 5.0] as follows:

$$\begin{gathered}\text{given } T > 0, \mathbf{u}_0 \in H \text{ and } \mathbf{f} \in L^2(J;H), \\ \text{find } \mathbf{u} \in L^\infty(J;H)\cap L^2(J;V), \text{ such that}, \forall \mathbf{v} \in V : \\ \frac{d}{dt}(\mathbf{u},\mathbf{v})_H + \nu((\mathbf{u},\mathbf{v})) + b(\mathbf{u},\mathbf{u},\mathbf{v}) = (\mathbf{f},\mathbf{v})_H .\end{gathered} \tag{1.4.59}$$

If $\mathbf{f}$ is square integrable but not with values in H, we can replace it by its Leray projection on H, such that $\mathbf{f}$ is always assumed to be in H. The pressure term disappears in the weak formulation because it is orthogonal to H [2, chapt. 5.0].

In Equation (1.4.59), the trilinear form b is defined by

$$b(\mathbf{u},\mathbf{v},\mathbf{w}) = \sum_{i,j=1}^{d} \int_D u_i \frac{\partial v_j}{\partial x_i} w_j \, d\mathbf{x} \, . \tag{1.4.60}$$

The trilinear form b is continuous on V and

$$\forall \mathbf{u},\mathbf{v},\mathbf{w} \in V : \quad b(\mathbf{u},\mathbf{v},\mathbf{v}) = 0 \quad \text{and} \quad b(\mathbf{u},\mathbf{v},\mathbf{w}) = -b(\mathbf{u},\mathbf{w},\mathbf{v}) \, . \tag{1.4.61}$$

Further, the trilinear form b induces, for fixed $\mathbf{u} \in V$, a bilinear operator $B : V \times V \to V^*$ defined by

$$_{V^*}\langle B(\mathbf{u},\mathbf{v}),\mathbf{w}\rangle_V = b(\mathbf{u},\mathbf{v},\mathbf{w}), \tag{1.4.62}$$

for all $\mathbf{u},\mathbf{v},\mathbf{w} \in V$ [1, sect. 3.2]. This corresponds to the same operator B that was used in the functional formulation (1.4.39) [2, chapt. 5.0].

Weak solutions of the Navier-Stokes equations are characterized, in the sense of Leray-Hopf, by the following definition [1, def. 3.1][3, def. 2.1].

Definition 1.4.1. On a time interval $J \subset \mathbb{R}$, a function $\mathbf{u} : J \mapsto H$ is called a *Leray–Hopf weak solution* of the Navier-Stokes equations (1.3.2) if

(i) $\mathbf{u} \in L^\infty(J;H) \cap L^2(J;V)$,

(ii) $(\partial_t \mathbf{u})(\cdot) \in L^{4/3}(J;V^*)$ for $d = 3$ or $(\partial_t \mathbf{u})(\cdot) \in L^2(J;V^*)$ for $d = 2$,

(iii) $t \mapsto \mathbf{u}(t)$ is weakly continuous (i.e. for every $\mathbf{v} \in H$, $t \mapsto (\mathbf{u}(t),\mathbf{v})_H$ is continuous from J to $\mathbb{R}$),

(iv) $\mathbf{u}$ satisfies (1.4.39),

(v) for almost all $t, t' \in J$, $\mathbf{u}$ satisfies the energy inequality

$$\frac{1}{2}|\mathbf{u}(t)|^2 + \nu \int_{t'}^{t} ||\mathbf{u}(s)||^2 \, ds \leq \frac{1}{2}|\mathbf{u}(t')|^2 + \int_{t'}^{t} (\mathbf{f}(s),\mathbf{u}(s)) \, ds. \tag{1.4.63}$$

We call Leray–Hopf solutions from now on simply *weak solutions* of the Navier–Stokes equation. For any $t_0 \in \mathbb{R}$ and for any $\mathbf{u}_0 \in H$, there exists at least one global weak solution in $[t_0,\infty)$ such that $\mathbf{u}(t_0) = \mathbf{u}_0$ in H. In space dimension $d = 2$, this solution is, moreover, unique [2, thm. II.7.1-4].

Solution operator

We denote by $S(t,0)$ the solution operator that maps $\mathbf{u}_0$ into $\mathbf{u}(t)$. The solution operator is well defined in space dimension $d = 2$ thanks to the uniqueness of weak solutions [2, chapt. 2.7]. It is, however, in general not a semigroup on H (solution operators are not associative), because $\mathbf{f}$ could be time dependent [1, sect. 3.2]. In space dimension $d = 3$, the definition of the solution operator is "more involved" [1, sect. 3.2], since in the presence of a time-dependent $\mathbf{f}$, only local uniqueness has been shown [2, thm. II.7.2].

In the next chapter, we introduce the concept of statistical solutions of the Navier-Stokes equations. But first, to conclude this chapter, we present an explicit expression for the eigenfunctions of the Stokes operator in the space-periodic case with vanishing space average in space dimension $d = 2$. As mentioned in Section 1.4.4, they constitute an orthonormal basis of H, and are therefore useful to expand data and solutions in H.

1.5 Eigenfunctions of the Stokes operator in the space-periodic case with vanishing space average

We present here another, explicit representation of the eigenfunctions of the Stokes operator

$$A\mathbf{u} = -P_L \triangle \mathbf{u} \tag{1.5.1}$$

for the space-periodic case with vanishing space-average in space dimension $d = 2$. We recall that in the periodic case, the Stokes operator reduces to

$$A\mathbf{u} = -\triangle \mathbf{u} \, . \tag{1.5.2}$$

In the following, we present the eigenfunctions of the Stokes operator together with a temporal evolution factor, sum up their properties, and show that these time-dependent eigenfunctions are actual solutions of the Navier-Stokes equations when the body force is conservative. This last property will be useful to help us calculate exact reference solutions for our numerical experiments.

1.5.1 Stokes eigenfunctions

The eigenfunctions of the Stokes operator are given (the eigenfunctions without the time dependent factor $A(t)$ were found in [16, append. A.1]) by

$$\mathbf{w}^{I}_{\kappa_1,\kappa_2}(\mathbf{x},t) = \begin{pmatrix} \kappa_2 \sin(\frac{2\pi}{L_1}\kappa_1 x_1)\sin(\frac{2\pi}{L_1}\kappa_2 x_2) \\ \kappa_1 \cos(\frac{2\pi}{L_1}\kappa_1 x_1)\cos(\frac{2\pi}{L_1}\kappa_2 x_2) \end{pmatrix} \frac{C_{\mathbf{w}}}{\sqrt{\kappa_1^2+\kappa_2^2}L_1} A(t) \, , \tag{1.5.3}$$

$$\mathbf{w}^{II}_{\kappa_1,\kappa_2}(\mathbf{x},t) = \begin{pmatrix} \kappa_2 \sin(\frac{2\pi}{L_1}\kappa_1 x_1)\cos(\frac{2\pi}{L_1}\kappa_2 x_2) \\ -\kappa_1 \cos(\frac{2\pi}{L_1}\kappa_1 x_1)\sin(\frac{2\pi}{L_1}\kappa_2 x_2) \end{pmatrix} \frac{C_{\mathbf{w}}}{\sqrt{\kappa_1^2+\kappa_2^2}L_1} A(t) \, , \tag{1.5.4}$$

$$\mathbf{w}^{III}_{\kappa_1,\kappa_2}(\mathbf{x},t) = \begin{pmatrix} \kappa_2 \cos(\frac{2\pi}{L_1}\kappa_1 x_1)\sin(\frac{2\pi}{L_1}\kappa_2 x_2) \\ -\kappa_1 \sin(\frac{2\pi}{L_1}\kappa_1 x_1)\cos(\frac{2\pi}{L_1}\kappa_2 x_2) \end{pmatrix} \frac{C_{\mathbf{w}}}{\sqrt{\kappa_1^2+\kappa_2^2}L_1} A(t) \, , \tag{1.5.5}$$

$$\mathbf{w}^{IV}_{\kappa_1,\kappa_2}(\mathbf{x},t) = \begin{pmatrix} \kappa_2 \cos(\frac{2\pi}{L_1}\kappa_1 x_1)\cos(\frac{2\pi}{L_1}\kappa_2 x_2) \\ \kappa_1 \sin(\frac{2\pi}{L_1}\kappa_1 x_1)\sin(\frac{2\pi}{L_1}\kappa_2 x_2) \end{pmatrix} \frac{C_{\mathbf{w}}}{\sqrt{\kappa_1^2+\kappa_2^2}L_1} A(t) \, , \tag{1.5.6}$$

with

$$A(t) \equiv e^{-\nu \frac{4\pi^2}{L_1^2}(\kappa_1^2+\kappa_2^2)t} \, , \tag{1.5.7}$$

where the time $t \in \mathbb{R}_{\geq 0}$, $\mathbf{x} = (x_1, x_2)$ in the periodic domain $D = (0, L_1) \times (0, L_1)$, $\kappa_1, \kappa_2 \in \mathbb{N}$, $C_{\mathbf{w}} \in \mathbb{R}$ is some prefactor (independent of x_1, x_2 and t), and $\nu > 0$ is the kinematic viscosity. The prefactor $C_{\mathbf{w}} < \infty$ can be chosen freely. For example, in some of our numerical experiments, $C_{\mathbf{w}}$ takes the values of a uniformly distributed random variable on the intervals $(0, 1)$ or $(-1, 1)$.

We remark that the eigenfunctions are also valid for $\kappa_1 \in \mathbb{N}$ and $\kappa_2 = 0$, and for $\kappa_1 = 0$ and $\kappa_2 \in \mathbb{N}$. In the former case, only $\mathbf{w}^{I}_{\kappa_1,0}$ and $\mathbf{w}^{III}_{\kappa_1,0}$ are not equal to zero. In the latter case, only $\mathbf{w}^{III}_{0,\kappa_2}$ and $\mathbf{w}^{IV}_{0,\kappa_2}$ are not equal to zero. For these non-trivial functions, all the properties presented in the next section remain valid, although for simplicity these functions are not listed there.

1.5.2 Properties

We show here that the functions presented above are effectively eigenfunctions of the Stokes operator and a set of orthogonal basis functions of the space of periodic divergence-free vector fields with finite kinetic energy and vanishing space-average, i.e. of the space

$$\dot{H}_{per} = \{\mathbf{v} \in L^2(D) : \nabla \cdot \mathbf{v} = 0, \int_D \mathbf{v}\, d\mathbf{x} = 0\} \,. \tag{1.5.8}$$

First, it is easy to show that the functions have a vanishing space average. It holds

$$\forall \mathcal{I} \in \{I, II, III, IV\},\, \forall \kappa_1, \kappa_2 \in \mathbb{N} : \int_D \mathbf{w}^{\mathcal{I}}_{\kappa_1,\kappa_2}\, d\mathbf{x} = 0 \,. \tag{1.5.9}$$

Their H-norm, which is the L^2-norm for the space $\dot{H}_{per}$, can also be easily calculated

$$\begin{aligned} &\forall \mathcal{I} \in \{I, II, III, IV\},\, \forall \kappa_1, \kappa_2 \in \mathbb{N},\, \forall t \geq 0 : \\ &\qquad \|\mathbf{w}^{\mathcal{I}}_{\kappa_1,\kappa_2}\|_H = \frac{1}{2} C_{\mathbf{w}} A(t) < \infty \,, \end{aligned} \tag{1.5.10}$$

as well as the orthogonality property

$$\begin{aligned} &\forall \mathcal{I}_1, \mathcal{I}_2 \in \{I, II, III, IV\},\, \forall \kappa_1, \kappa_2, \kappa_1', \kappa_2' \in \mathbb{N} \text{ s.t. } \mathcal{I}_1 \neq \mathcal{I}_2 \\ &\text{and/or } (\kappa_1, \kappa_2) \neq (\kappa_1', \kappa_2') : (\mathbf{w}^{\mathcal{I}_1}_{\kappa_1,\kappa_2}, \mathbf{w}^{\mathcal{I}_2}_{\kappa_1',\kappa_2'})_H = 0 \,. \end{aligned} \tag{1.5.11}$$

Moreover, it is easy to show that they satisfy the continuity equation, i.e.

$$\forall \mathcal{I} \in \{I, II, III, IV\},\, \forall \kappa_1, \kappa_2 \in \mathbb{N} : \nabla \cdot \mathbf{w}^{\mathcal{I}}_{\kappa_1,\kappa_2} = 0 \,. \tag{1.5.12}$$

Thus, the functions $\mathbf{w}^{\mathcal{I}}_{\kappa_1,\kappa_2}$ are in $\dot{H}_{per}$ and mutually orthogonal, and form an orthogonal basis of $\dot{H}_{per}$. We see from (1.5.10) and (1.5.7) that by choosing $C_{\mathbf{w}} = 2$ at time $t = 0$, they form an *orthonormal* basis of $\dot{H}_{per}$. As mentioned in Section 1.4.4, the Stokes operator reduces to the negative Laplacian for the space $\dot{H}_{per}$. By applying the negative Laplacian to the functions $\mathbf{w}^{\mathcal{I}}_{\kappa_1,\kappa_2}$, we get

$$\begin{aligned} &\forall \mathcal{I} \in \{I, II, III, IV\},\, \forall \kappa_1, \kappa_2 \in \mathbb{N} : \\ &-\Delta_{\mathbf{x}} \mathbf{w}^{\mathcal{I}}_{\kappa_1,\kappa_2} = \frac{4\pi^2}{L_1^2}(\kappa_1^2 + \kappa_2^2)\mathbf{w}^{\mathcal{I}}_{\kappa_1,\kappa_2} \,, \end{aligned} \tag{1.5.13}$$

which shows that the functions $\mathbf{w}^{\mathcal{I}}_{\kappa_1,\kappa_2}$ are eigenfunctions of the Stokes operator with corresponding eigenvalues

$$\lambda_{\kappa_1,\kappa_2} = \frac{4\pi^2}{L_1^2}(\kappa_1^2 + \kappa_2^2) \,. \tag{1.5.14}$$

We remark finally that the derivative in time and the negative Laplacian of these functions cancel out, i.e.

$$\forall \mathcal{I} \in \{I, II, III, IV\},\, \forall \kappa_1, \kappa_2 \in \mathbb{N} : \partial_t \mathbf{w}^{\mathcal{I}}_{\kappa_1,\kappa_2} - \nu \triangle \mathbf{w}^{\mathcal{I}}_{\kappa_1,\kappa_2} = 0 \,. \tag{1.5.15}$$

Thus, the momentum equation

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \triangle \mathbf{u} + \mathbf{f} \tag{1.5.16}$$

reduces to

$$\frac{1}{\rho}\nabla p = \mathbf{f} - (\mathbf{u} \cdot \nabla)\mathbf{u} \tag{1.5.17}$$

for the functions $\mathbf{u} = \mathbf{w}^{\mathcal{I}}_{\kappa_1,\kappa_2}$. This last property will be useful in the next section, where we show that the functions $\mathbf{w}^{\mathcal{I}}_{\kappa_1,\kappa_2}$ are solutions of the Navier-Stokes equations.

1.5.3 Exact solution when f is conservative

We assume that we can write the forcing term as a conservative body force, i.e.

$$\mathbf{f}(\mathbf{x},t) = -\nabla\psi(\mathbf{x},t)\ . \tag{1.5.18}$$

As an example, we have $\psi = gx_2$ (constant gravitational field, g being the gravitational acceleration) and the trivial case $\psi = 0$, $\mathbf{f} = \mathbf{0}$. Then we can define the modified pressure field

$$\widetilde{p} = \frac{1}{\rho}p + \psi\ , \tag{1.5.19}$$

and the momentum equation (1.5.16) reduces to

$$\partial_t\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla\widetilde{p} + \nu\triangle\mathbf{u}\ . \tag{1.5.20}$$

That is, the body force has no effect on the velocity field and on the modified pressure field. We recover the pressure field p by

$$p(\mathbf{x},t) = \rho\widetilde{p}(\mathbf{x},t) - \rho\psi(\mathbf{x},t)\ . \tag{1.5.21}$$

By using Equation (1.5.15), for the basis functions $\mathbf{w}^{\mathcal{I}}_{\kappa_1,\kappa_2}$, the momentum equation (1.5.20) reduces then to

$$\nabla\widetilde{p} = -(\mathbf{w}^{\mathcal{I}}_{\boldsymbol{\kappa_1},\boldsymbol{\kappa_2}}\cdot\nabla)\mathbf{w}^{\mathcal{I}}_{\boldsymbol{\kappa_1},\boldsymbol{\kappa_2}}\ . \tag{1.5.22}$$

We prove next that the basis functions $\mathbf{w}^{\mathcal{I}}_{\kappa_1,\kappa_2}$ solve (1.5.22) and are thus solutions of the Navier-Stokes equations with a conservative body force. From the definitions (1.5.3), (1.5.4), (1.5.5) and (1.5.6), this means that they don't change during their evolution in time, except that they are damped by the factor $A(t)$ defined in (1.5.7). We show here only the case $\mathcal{I} = I$, but the proof is similar for the other basis functions.

Let us define $\mathbf{u}(\mathbf{x},t) \equiv \mathbf{w}^{I}_{\kappa_1,\kappa_2}(\mathbf{x},t)$, and use the following abbreviations to facilitate the notation:

$$\begin{aligned} &u_1 \equiv \mathbf{u}(\mathbf{x},t)_1, u_2 \equiv \mathbf{u}(\mathbf{x},t)_2, \boldsymbol{\kappa} \equiv (\kappa_1,\kappa_2),\\ &s_1 \equiv \sin(\frac{2\pi\kappa_1}{L_1}x_1), s_2 \equiv \sin(\frac{2\pi\kappa_2}{L_1}x_2),\\ &c_1 \equiv \cos(\frac{2\pi\kappa_1}{L_1}x_1), c_2 \equiv \cos(\frac{2\pi\kappa_2}{L_1}x_2)\ . \end{aligned} \tag{1.5.23}$$

Then it follows

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \kappa_2 s_1 s_2 \\ \kappa_1 c_1 c_2 \end{pmatrix} \frac{C_{\mathbf{w}}}{\|\boldsymbol{\kappa}\|_2 L_1} A(t)\ , \tag{1.5.24}$$

$$\partial_{x_1}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \kappa_2 c_1 s_2 \\ -\kappa_1 s_1 c_2 \end{pmatrix} \frac{2\pi\kappa_1}{L_1}\frac{C_{\mathbf{w}}}{\|\boldsymbol{\kappa}\|_2 L_1} A(t)\ , \tag{1.5.25}$$

and

$$\partial_{x_2}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \kappa_2 s_1 c_2 \\ -\kappa_1 c_1 s_2 \end{pmatrix} \frac{2\pi\kappa_2}{L_1}\frac{C_{\mathbf{w}}}{\|\boldsymbol{\kappa}\|_2 L_1} A(t)\ . \tag{1.5.26}$$

As mentioned before, the momentum equation results in

$$(\mathbf{u}(\mathbf{x},t)\cdot\nabla)\mathbf{u}(\mathbf{x},t) = -\nabla\widetilde{p}(\mathbf{x},t)\ . \tag{1.5.27}$$

We calculate

$$\begin{aligned}
-\partial_{x_1}\widetilde{p} &= u_1\partial_{x_1}u_1 + u_2\partial_{x_2}u_1 \\
&= [\frac{2\pi}{L_1}\kappa_1\kappa_2^2 s_1c_1s_2^2 + \frac{2\pi}{L_1}\kappa_1\kappa_2^2 s_1c_1c_2^2]\big(\frac{C_{\mathbf{w}}}{\|\boldsymbol{\kappa}\|_2 L_1}A(t)\big)^2 \\
&= [\kappa_2^2\frac{2\pi\kappa_1}{L_1}\underbrace{s_1c_1}_{\frac{1}{2}\sin(2\frac{2\pi\kappa_1}{L_1}x_1)}\underbrace{(s_2^2+c_2^2)}_{=1}]\big(\frac{C_{\mathbf{w}}}{\|\boldsymbol{\kappa}\|_2 L_1}A(t)\big)^2 \\
&= \frac{1}{4}\kappa_2^2\frac{4\pi\kappa_1}{L_1}\sin(\frac{4\pi\kappa_1}{L_1}x_1)\big(\frac{C_{\mathbf{w}}}{\|\boldsymbol{\kappa}\|_2 L_1}A(t)\big)^2 ,
\end{aligned} \tag{1.5.28}$$

and

$$\begin{aligned}
-\partial_{x_2}\widetilde{p} &= u_1\partial_{x_1}u_2 + u_2\partial_{x_2}u_2 \\
&= [-\frac{2\pi}{L_1}\kappa_1^2\kappa_2 s_1^2s_2c_2 - \frac{2\pi}{L_1}\kappa_1\kappa_2^2 c_1^2s_2c_2]\big(\frac{C_{\mathbf{w}}}{\|\boldsymbol{\kappa}\|_2 L_1}A(t)\big)^2 \\
&= [-\kappa_1^2\frac{2\pi\kappa_2}{L_1}\underbrace{s_2c_2}_{\frac{1}{2}\sin(2\frac{2\pi\kappa_2}{L_1}x_2)}\underbrace{(s_1^2+c_1^2)}_{=1}]\big(\frac{C_{\mathbf{w}}}{\|\boldsymbol{\kappa}\|_2 L_1}A(t)\big)^2 \\
&= -\frac{1}{4}\kappa_1^2\frac{4\pi\kappa_2}{L_1}\sin(\frac{4\pi\kappa_2}{L_1}x_2)\big(\frac{C_{\mathbf{w}}}{\|\boldsymbol{\kappa}\|_2 L_1}A(t)\big)^2 .
\end{aligned} \tag{1.5.29}$$

Thus, we showed that the basis function $\mathbf{w}^I_{\kappa_1,\kappa_2}(\mathbf{x},t)$ solves the Navier-Stokes equations with a conservative body force, and the modified pressure is given by

$$\widetilde{p}^I_{\kappa_1,\kappa_2}(\mathbf{x},t) = C_{\widetilde{p}} + \frac{1}{4}\big[\frac{C_{\mathbf{w}}A(t)}{\|\boldsymbol{\kappa}\|_2 L_1}\big]^2\big[\kappa_2^2\cos(\frac{4\pi\kappa_1}{L_1}x_1) - \kappa_1^2\cos(\frac{4\pi\kappa_2}{L_1}x_2)\big] , \tag{1.5.30}$$

where $C_{\widetilde{p}} \in \mathbb{R}$ is an arbitrary constant. The pressure $p^I_{\kappa_1,\kappa_2}$ is then given by

$$p^I_{\kappa_1,\kappa_2}(\mathbf{x},t) = C_p + \frac{\rho}{4}\big[\frac{C_{\mathbf{w}}A(t)}{\|\boldsymbol{\kappa}\|_2 L_1}\big]^2\big[\kappa_2^2\cos(\frac{4\pi\kappa_1}{L_1}x_1) - \kappa_1^2\cos(\frac{4\pi\kappa_2}{L_1}x_2)\big] - \rho\psi(\mathbf{x},t) , \tag{1.5.31}$$

with $C_p = \rho C_{\widetilde{p}}$.

Analogously, we get for $\mathbf{w}^{II}_{\kappa_1,\kappa_2}(\mathbf{x},t)$, $\mathbf{w}^{III}_{\kappa_1,\kappa_2}(\mathbf{x},t)$ and $\mathbf{w}^{IV}_{\kappa_1,\kappa_2}(\mathbf{x},t)$:

$$\begin{aligned}
p^{II}_{\kappa_1,\kappa_2}(\mathbf{x},t) &= C_p \\
&\quad + \frac{\rho}{4}\big[\frac{C_{\mathbf{w}}A(t)}{\|\boldsymbol{\kappa}\|_2 L_1}\big]^2\big[\quad \kappa_2^2\cos(\frac{4\pi\kappa_1}{L_1}x_1) + \kappa_1^2\cos(\frac{4\pi\kappa_2}{L_1}x_2)\big] - \rho\psi(\mathbf{x},t) ,
\end{aligned} \tag{1.5.32}$$

$$\begin{aligned}
p^{III}_{\kappa_1,\kappa_2}(\mathbf{x},t) &= C_p \\
&\quad + \frac{\rho}{4}\big[\frac{C_{\mathbf{w}}A(t)}{\|\boldsymbol{\kappa}\|_2 L_1}\big]^2\big[-\kappa_2^2\cos(\frac{4\pi\kappa_1}{L_1}x_1) - \kappa_1^2\cos(\frac{4\pi\kappa_2}{L_1}x_2)\big] - \rho\psi(\mathbf{x},t) ,
\end{aligned} \tag{1.5.33}$$

and

$$\begin{aligned}
p^{IV}_{\kappa_1,\kappa_2}(\mathbf{x},t) &= C_p \\
&\quad + \frac{\rho}{4}\big[\frac{C_{\mathbf{w}}A(t)}{\|\boldsymbol{\kappa}\|_2 L_1}\big]^2\big[-\kappa_2^2\cos(\frac{4\pi\kappa_1}{L_1}x_1) + \kappa_1^2\cos(\frac{4\pi\kappa_2}{L_1}x_2)\big] - \rho\psi(\mathbf{x},t) .
\end{aligned} \tag{1.5.34}$$

Next we show that the sum of all four basis functions also solves the Navier-Stokes equations (1.5.20), assuming they all have the same wave numbers κ_1 and κ_2. This property will be useful for calculating exact reference solutions to our numerical experiments.

Let us define $\mathbf{u}(\mathbf{x},t) \equiv \mathbf{w}^I_{\kappa_1,\kappa_2}(\mathbf{x},t) + \mathbf{w}^{II}_{\kappa_1,\kappa_2}(\mathbf{x},t) + \mathbf{w}^{III}_{\kappa_1,\kappa_2}(\mathbf{x},t) + \mathbf{w}^{IV}_{\kappa_1,\kappa_2}(\mathbf{x},t)$. Then, using the same abbreviations as in (1.5.23), it follows

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \kappa_2(s_1s_2 + s_1c_2 + c_1s_2 + c_1c_2) \\ \kappa_1(c_1c_2 - c_1s_2 - s_1c_2 + s_1s_2) \end{pmatrix} \frac{C_{\mathbf{w}}}{\|\boldsymbol{\kappa}\|_2 L_1} A(t) \,, \tag{1.5.35}$$

$$\partial_{x_1} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \kappa_2(c_1s_2 + c_1c_2 - s_1s_2 - s_1c_2) \\ \kappa_1(-s_1c_2 + s_1s_2 - c_1c_2 + c_1s_2) \end{pmatrix} \frac{2\pi\kappa_1}{L_1} \frac{C_{\mathbf{w}}}{\|\boldsymbol{\kappa}\|_2 L_1} A(t) \,, \tag{1.5.36}$$

and

$$\partial_{x_2} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \kappa_2(s_1c_2 - s_1s_2 + c_1c_2 - c_1s_2) \\ \kappa_1(-c_1s_2 - c_1c_2 + s_1s_2 + s_1c_2) \end{pmatrix} \frac{2\pi\kappa_2}{L_1} \frac{C_{\mathbf{w}}}{\|\boldsymbol{\kappa}\|_2 L_1} A(t) \,. \tag{1.5.37}$$

We calculate

$$\begin{aligned} u_1\partial_{x_1}u_1 = \kappa_2^2\frac{2\pi\kappa_1}{L_1}&(s_1c_1s_2^2 + s_1s_2c_1c_2 - s_1^2s_2^2 - s_1^2s_2c_2 \\ &+ s_1c_2c_1s_2 + s_1c_1c_2^2 - s_1^2c_2s_2 - s_1^2c_2^2 \\ &+ c_1^2s_2^2 + c_1^2s_2c_2 - c_1s_1s_2^2 - c_1s_2s_1c_2 \\ &+ c_1^2c_2s_2 + c_1^2c_2^2 - c_1s_1c_2s_2 - c_1s_1c_2^2)\big[\frac{C_{\mathbf{w}}A(t)}{\|\boldsymbol{\kappa}\|_2 L_1}\big]^2 \\ = \kappa_2^2\frac{2\pi\kappa_1}{L_1}&(-s_1^2s_2^2 - s_1^2c_2^2 + c_1^2s_2^2 + c_1^2c_2^2 - 2s_1^2s_2c_2 + 2c_1^2s_2c_2)\big[\frac{C_{\mathbf{w}}A(t)}{\|\boldsymbol{\kappa}\|_2 L_1}\big]^2 \\ = \kappa_2^2\frac{2\pi\kappa_1}{L_1}&(-s_1^2\underbrace{(s_2^2+c_2^2)}_{=1} + c_1^2\underbrace{(s_2^2+c_2^2)}_{=1} + 2s_2c_2(c_1^2 - s_1^2))\big[\frac{C_{\mathbf{w}}A(t)}{\|\boldsymbol{\kappa}\|_2 L_1}\big]^2 \\ = \kappa_2^2\frac{2\pi\kappa_1}{L_1}&(c_1^2 - s_1^2)(1 + 2s_2c_2)\big[\frac{C_{\mathbf{w}}A(t)}{\|\boldsymbol{\kappa}\|_2 L_1}\big]^2 \,. \end{aligned} \tag{1.5.38}$$

Analogously, we have

$$u_2\partial_{x_2}u_1 = \kappa_2^2\frac{2\pi\kappa_1}{L_1}(c_1^2 - s_1^2)(1 - 2s_2c_2)\big[\frac{C_{\mathbf{w}}A(t)}{\|\boldsymbol{\kappa}\|_2 L_1}\big]^2 \,. \tag{1.5.39}$$

From the linearity of the Laplacian operator and of the partial derivative in time, we have that here also the momentum equation (1.5.20) results in

$$(\mathbf{u}(\mathbf{x},t)\cdot\nabla)\mathbf{u}(\mathbf{x},t) = -\nabla\widetilde{p}(\mathbf{x},t) \,. \tag{1.5.40}$$

This gives

$$\begin{aligned} -\partial_{x_1}\widetilde{p} &= u_1\partial_{x_1}u_1 + u_2\partial_{x_2}u_1 \\ &= \kappa_2^2\frac{2\pi\kappa_1}{L_1}2\underbrace{(c_1^2 - s_1^2)}_{\cos(2\frac{2\pi}{L_1}\kappa_1 x_1)}\big[\frac{C_{\mathbf{w}}}{\|\boldsymbol{\kappa}\|_2 L_1}A(t)\big]^2 \\ &= \kappa_2^2\frac{4\pi\kappa_1}{L_1}\cos(\frac{4\pi\kappa_1}{L_1}x_1)\big[\frac{C_{\mathbf{w}}}{\|\boldsymbol{\kappa}\|_2 L_1}A(t)\big]^2 \,. \end{aligned} \tag{1.5.41}$$

Similarly, we have

$$u_1\partial_{x_1}u_2 = \kappa_1^2\frac{2\pi\kappa_2}{L_1}(s_2^2 - c_2^2)(1 + 2s_1c_1)\big[\frac{C_{\mathbf{w}}}{\|\boldsymbol{\kappa}\|_2 L_1}A(t)\big]^2 \,, \tag{1.5.42}$$

and

$$u_2\partial_{x_2}u_2 = \kappa_1^2\frac{2\pi\kappa_2}{L_1}(s_2^2-c_2^2)(1-2s_1c_1)\big[\frac{C_{\mathbf{w}}}{\|\boldsymbol{\kappa}\|_2 L_1}A(t)\big]^2 . \tag{1.5.43}$$

This yields

$$\begin{aligned} -\partial_{x_2}\widetilde{p} &= u_1\partial_{x_1}u_2 + u_2\partial_{x_2}u_2 \\ &= \kappa_1^2\frac{2\pi\kappa_2}{L_1}2\underbrace{(s_2^2-c_2^2)}_{-\cos(2\frac{2\pi}{L_1}\kappa_2 x_2)}\big[\frac{C_{\mathbf{w}}}{\|\boldsymbol{\kappa}\|_2 L_1}A(t)\big]^2 \\ &= -\kappa_1^2\frac{4\pi\kappa_2}{L_1}\cos(\frac{4\pi\kappa_2}{L_1}x_2)\big[\frac{C_{\mathbf{w}}}{\|\boldsymbol{\kappa}\|_2 L_1}A(t)\big]^2 . \end{aligned} \tag{1.5.44}$$

Thus, we showed that the sum of the basis functions $\mathbf{w}^I_{\kappa_1,\kappa_2}(\mathbf{x},t)+\mathbf{w}^{II}_{\kappa_1,\kappa_2}(\mathbf{x},t)+\mathbf{w}^{III}_{\kappa_1,\kappa_2}(\mathbf{x},t)+\mathbf{w}^{IV}_{\kappa_1,\kappa_2}(\mathbf{x},t)$ (with *same* wave number vector $\boldsymbol{\kappa}=(\kappa_1,\kappa_2)$) solves the Navier-Stokes equations with a conservative body force, and the modified pressure is given by

$$\widetilde{p}_{\kappa_1,\kappa_2}(\mathbf{x},t) = C_{\widetilde{p}} + \big[\frac{C_{\mathbf{w}}A(t)}{\|\boldsymbol{\kappa}\|_2 L_1}\big]^2\big[-\kappa_2^2\sin(\frac{4\pi\kappa_1}{L_1}x_1)+\kappa_1^2\sin(\frac{4\pi\kappa_2}{L_1}x_2)\big] , \tag{1.5.45}$$

where $C_{\widetilde{p}}\in\mathbb{R}$ is an arbitrary constant. The pressure is then given by

$$p_{\kappa_1,\kappa_2}(\mathbf{x},t) = C_p + \rho\big[\frac{C_{\mathbf{w}}A(t)}{\|\boldsymbol{\kappa}\|_2 L_1}\big]^2\big[-\kappa_2^2\sin(\frac{4\pi\kappa_1}{L_1}x_1)+\kappa_1^2\sin(\frac{4\pi\kappa_2}{L_1}x_2)\big]-\rho\psi(\mathbf{x},t) , \tag{1.5.46}$$

with $C_p=\rho C_{\widetilde{p}}$.

We conclude this section by showing in Figures 1.1, 1.2, 1.3 and 1.4 the plots of the basis functions $\mathbf{w}^{\mathcal{I}}_{\kappa_1,\kappa_2}$ on the domain $D=(0,1)\times(0,1)$ at time $t=0$ and with $C_{\mathbf{w}}=1$, for the wave number vectors $\boldsymbol{\kappa}=(1,1)$ and $\boldsymbol{\kappa}=(1,2)$.

1.5.4 Plots

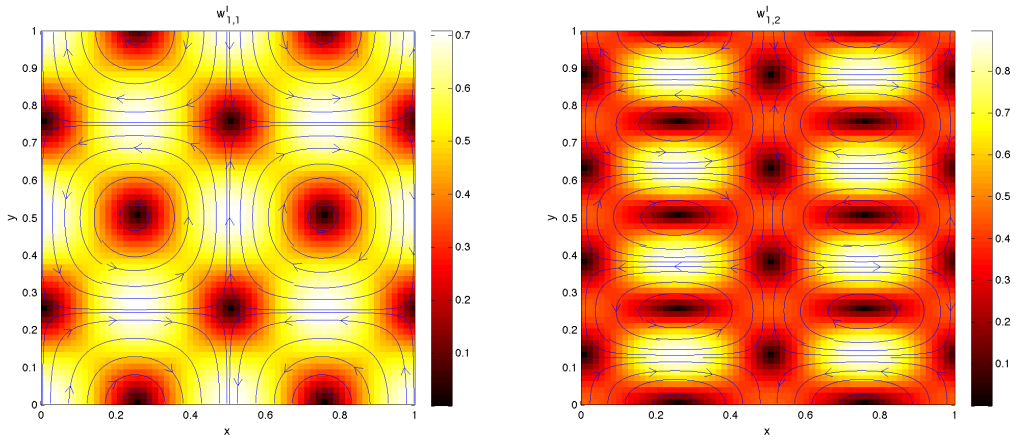


Figure 1.1: Stokes eigenfunctions $\mathbf{w}_{1,1}^I$ and $\mathbf{w}_{1,2}^I$ on $D = (0,1) \times (0,1)$ at time $t = 0$ and with $C_{\mathbf{w}} = 1$. Figure generated with MATLAB.

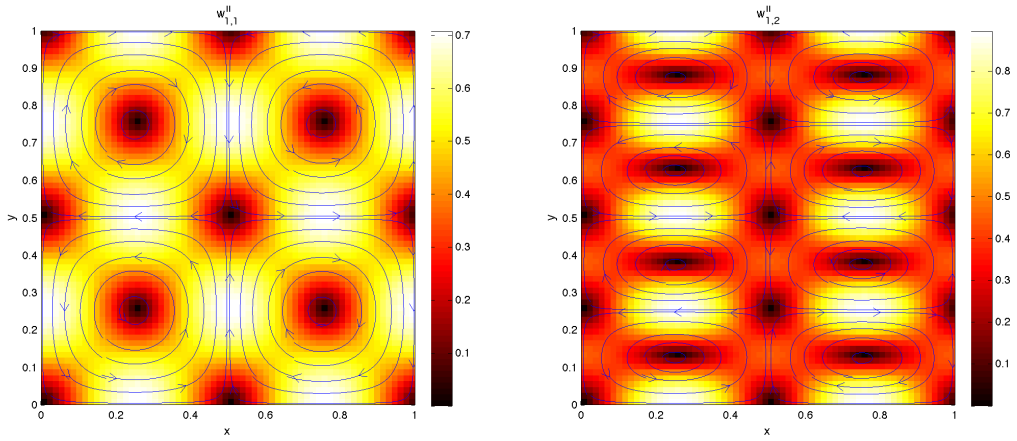


Figure 1.2: Stokes eigenfunctions $\mathbf{w}_{1,1}^{II}$ and $\mathbf{w}_{1,2}^{II}$ on $D = (0,1) \times (0,1)$ at time $t = 0$ and with $C_{\mathbf{w}} = 1$. Figure generated with MATLAB.

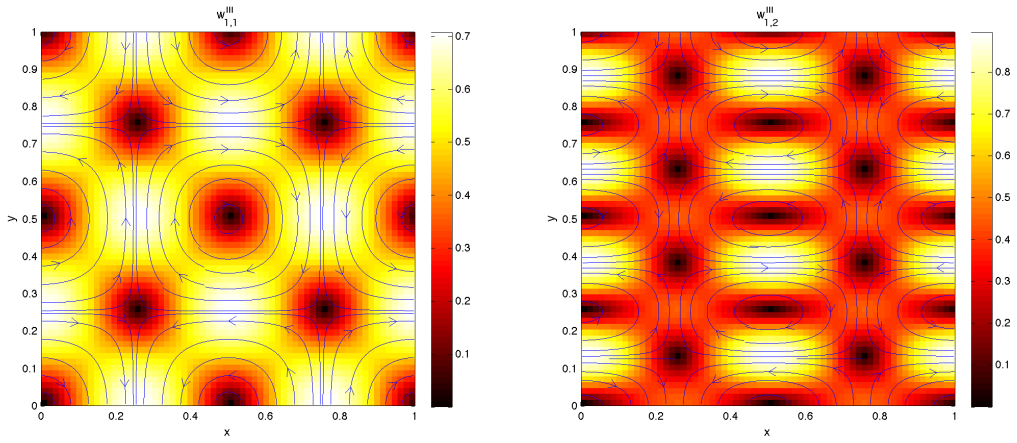


Figure 1.3: Stokes eigenfunctions $\mathbf{w}_{1,1}^{III}$ and $\mathbf{w}_{1,2}^{III}$ on $D = (0,1) \times (0,1)$ at time $t = 0$ and with $C_{\mathbf{w}} = 1$. Figure generated with MATLAB.

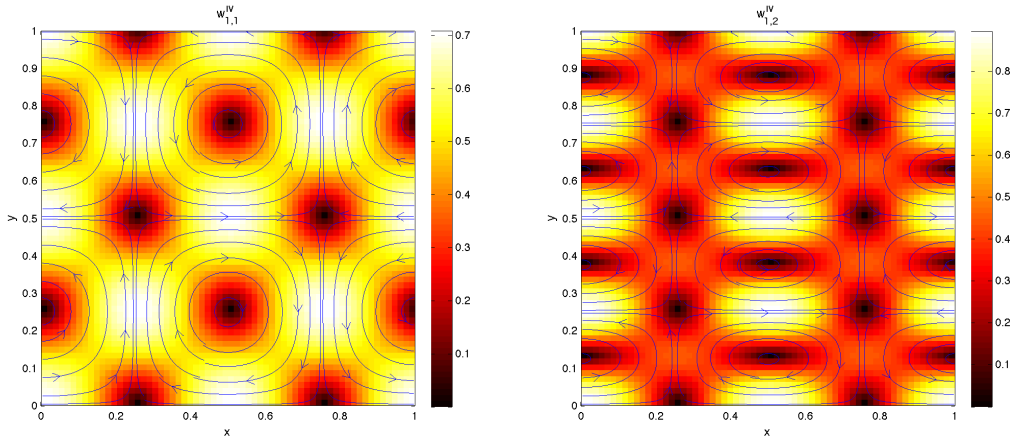


Figure 1.4: Stokes eigenfunctions $\mathbf{w}_{1,1}^{IV}$ and $\mathbf{w}_{1,1}^{IV}$ on $D = (0,1) \times (0,1)$ at time $t = 0$ and with $C_{\mathbf{w}} = 1$. Figure generated with MATLAB.

Chapter 2

Statistical solutions of the Navier-Stokes equations

As stated in [2, chapt. 5.0], it is "commonly accepted that turbulent flows are necessarily statistical in nature". However, the Navier-Stokes equations (1.3.2) are not a stochastic PDE, such that the statistical nature of turbulent flows should be attainable by other means. In the following, we consider as initial data an ensemble of velocity fields, and let all these velocities evolve according to the Navier-Stokes equations and get an ensemble of solutions. Uncertainty is then given by some probability distribution on the initial ensemble. This is in practice sufficient for the statistics on the ensemble of solutions to emulate "complex turbulent flows" [2, chapt. 5.0].

2.1 Probability distribution on the initial data

In the dynamical systems viewpoint of the Navier-Stokes equations presented in Chapter 1, we were interested in finding a (unique in space dimension $d = 2$) weak solution $\mathbf{u}(t)$ of (1.3.2) that solves the system

$$\mathbf{u}'(t) = \mathbf{F}(t, \mathbf{u}(t)), \quad \text{with} \quad \mathbf{F}(t, \mathbf{u}) = \mathbf{f}(t) - \nu A\mathbf{u} - B(\mathbf{u}), \tag{2.1.1}$$

given an initial (deterministic) velocity field $\mathbf{u}_0 \in H$, and denoted by $S(t, 0)$ the corresponding solution operator (well-defined in space dimension $d = 2$) in H that maps $\mathbf{u}_0$ into $\mathbf{u}(t)$. Now we try to describe the evolution if the initial velocity field is random.

More precisely, we consider an ensemble of initial velocity fields described by a given probability distribution μ_0 on the space H. Then the ensemble of solutions at some later time t will be described by (possibly) another probability distribution μ_t [2, chapt. 4.0], and we have a time-dependent family of measures $\mu = (\mu_t, t \geq 0)$ on H that are given by

$$\mu_t(E) = \mu_0(S(t, 0)^{-1}E), \tag{2.1.2}$$

for all measurable (sub-)ensembles of initial velocities $E \subset H$ [2, chapt. 5.1].

In general, the initial distribution is defined on an underlying (complete) probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and is assumed to be given as an image measure under a random variable $\mathbf{X}$ from the measurable space $(\Omega, \mathcal{F})$ into the measurable space $(H, \mathcal{B}(H))$, where $\mathcal{B}(H)$ is the Borel σ-algebra on H, i.e.

$$\mathbf{X} : \begin{cases} \Omega \to H \\ \omega \mapsto \mathbf{u}_0 \end{cases} \tag{2.1.3}$$

and

$$\mu_0(E') = \mathbb{P}\big(\{\omega \in \Omega : \mathbf{X}(\omega) \in E'\}\big) \tag{2.1.4}$$

for all $E' \in \mathcal{B}(H)$ [1, sect. 3.4].

Before continuing the discussion, we present the definitions of two norms we are using in the thesis. These definitions can be found in [1, sect. 2].

Definitions

For a random variable $X : \Omega \to B$ taking values in a Banach space B, the *expectation* of X is given by

$$\mathbb{E}(X) = \int_\Omega X \, d\mathbb{P} \, . \tag{2.1.5}$$

The expectation $\mathbb{E}(X)$ is defined for $X \in L^1(\Omega; B)$, where $L^p(\Omega; B)$, $1 \leq p < \infty$ (with a modification for $p = \infty$), denotes the space of p-summable random variables taking values in B, and is equipped with the norm

$$\|X\|_{L^p(\Omega;B)} := \begin{cases} \big(\mathbb{E}(\|X\|_B^p)\big)^{1/p}, & \text{for } 1 \leq p < \infty, \\ \operatorname*{ess\,sup}_{\omega \in \Omega} \|X(\omega)\|_B, & \text{for } p = \infty. \end{cases} \tag{2.1.6}$$

Similarly, the space of strongly measurable functions f taking values in B is denoted by $L^p(\bar{J}; B)$, and is equipped with the norm

$$\|f\|_{L^p(\bar{J};B)} := \begin{cases} \big(\int_0^T \|f(t)\|_B^p \, dt\big)^{1/p}, & \text{for } 1 \leq p < \infty, \\ \operatorname{ess\,sup}_{t \in (0,T)} \|f(t)\|_B, & \text{for } p = \infty. \end{cases} \tag{2.1.7}$$

For the next section, we follow the description in [2, chapt. 5].

2.2 Generalized moments

For any time $t \geq 0$, one can extract statistical information from the probability distribution μ_t through the generalized moment

$$\mathbb{E}_{\mu_t}(\Phi) = \int_H \Phi(\mathbf{v}) \, d\mu_t(\mathbf{v}) \tag{2.2.1}$$

for a μ_t-integrable function Φ on H. We will call this generalized moment also *ensemble average*. Heuristically, the more moments we have, the more we know about the probability distribution.

The simplest moments are the linear ones corresponding to the average velocity [2, chapt. 5.0]

$$\int_H v_i \, d\mu_t(\mathbf{v}), \quad i = 1, 2, 3 \, . \tag{2.2.2}$$

We may also consider nonlinear moments [2, chapt. 5.0] such as

$$\int_H v_{i_1} \dots v_{i_k} \, d\mu_t(\mathbf{v}) \, , \tag{2.2.3}$$

for example to calculate covariances. We may otherwise be interested in some scalar bulk property, that can be extracted by taking the inner product $(\cdot,\cdot)_H$ of the velocity $\mathbf{v} \in H$ with a given function $\mathbf{g} \in H$. The ensemble average of this bulk property is then

$$\int_H (\mathbf{v}, \mathbf{g})_H \, d\mu_t(\mathbf{v}) \, . \tag{2.2.4}$$

For example [2, chapt. 5.0], the ensemble average of the averaged first component of the velocity field in a ball of radius ε centered at $\mathbf{x}_0$ can be obtained with $\mathbf{g} = \frac{1}{\mathrm{Vol}(B_\varepsilon(\mathbf{x}_0))} \mathbf{1}_{B_\varepsilon(\mathbf{x}_0)} \mathbf{e}_1$, such that $(\mathbf{v}, \mathbf{g})_H = \frac{1}{\mathrm{Vol}(B_\varepsilon(\mathbf{x}_0))} \int_{B_\varepsilon(\mathbf{x}_0)} v_1(\mathbf{x}) \, d\mathbf{x}$.
Or more generally, we may consider the ensemble average of a function of several bulk properties

$$\int_H \phi\big((\mathbf{v}, \mathbf{g}_1)_H, \ldots, (\mathbf{v}, \mathbf{g}_1)_H\big) \, d\mu_t(\mathbf{v}) \, . \tag{2.2.5}$$

Let then Φ be a real-valued μ_t-integrable function on H. It holds [2, chapt. 5.1]

$$\int_H \Phi(\mathbf{v}) \, d\mu_t(\mathbf{v}) = \int_H \Phi(S(t,0)\mathbf{v}) \, d\mu_0(\mathbf{v}) \, , \tag{2.2.6}$$

such that the evolution in time is given by

$$\frac{d}{dt} \int_H \Phi(\mathbf{v}) \, d\mu_t(\mathbf{v}) = \int_H \frac{d}{dt} \Phi(S(t,0)\mathbf{v}) \, d\mu_0(\mathbf{v}) \, . \tag{2.2.7}$$

The time derivative of $\Phi(S(t,0)\mathbf{v})$ can be computed by the chain differentiation rule [2, chapt. 5.1]

$$\frac{d}{dt} \Phi(S(t,0)\mathbf{v}) = \big(\frac{d}{dt} S(t,0)\mathbf{v}, \Phi'(S(t,0)\mathbf{v})\big)_H = \big(\mathbf{F}(t, S(t,0)\mathbf{v}), \Phi'(S(t,0)\mathbf{v})\big)_H \, , \tag{2.2.8}$$

with $\mathbf{F}$ as in Equation (2.1.1). Thus, the evolution of statistical moments of the flow in time is given by

$$\begin{aligned} \frac{d}{dt} \int_H \Phi(\mathbf{v}) \, d\mu_t(\mathbf{v}) &= \int_H (\mathbf{F}(t, S(t,0)\mathbf{v}), \Phi'(S(t,0)\mathbf{v}))_H \, d\mu_0(\mathbf{v}) \\ &= \int_H (\mathbf{F}(t, \mathbf{v}), \Phi'(\mathbf{v}))_H \, d\mu_t(\mathbf{v}), \end{aligned} \tag{2.2.9}$$

for suitable testfunctionals Φ. The expression

$$\frac{d}{dt} \int_H \Phi(\mathbf{v}) \, d\mu_t(\mathbf{v}) = \int_H (\mathbf{F}(t, \mathbf{v}), \Phi'(\mathbf{v}))_H \, d\mu_t(\mathbf{v}) \tag{2.2.10}$$

is meaningful even if the solution operator is not defined, as in the general case for $d = 3$ [2, chapt. 5.1]. A suitable class of testfunctionals for (2.2.10) is given by the following definition.

Definition 2.2.1. Let $\mathcal{C}$ be the space of cylindrical test functionals Φ on H which are real-valued and depend only on a finite number of components of $\mathbf{v} \in H$, i.e. for $k < \infty$

$$\Phi(\mathbf{v}) = \phi((\mathbf{v}, \mathbf{g}_1)_H, \ldots, (\mathbf{v}, \mathbf{g}_k)_H), \tag{2.2.11}$$

where ϕ is a compactly supported C^1 scalar function on $\mathbb{R}^k$ and $\mathbf{g}_1, \ldots, \mathbf{g}_k \in V$.

For $\Phi \in \mathcal{C}$ we denote by Φ' its differential in H, which is given by

$$\Phi'(\mathbf{v}) = \sum_{i=1}^{k} \partial_i \phi((\mathbf{v}, \mathbf{g}_1)_H, \dots, (\mathbf{v}, \mathbf{g}_k)_H)\mathbf{g}_i \tag{2.2.12}$$

for the no-slip case and by

$$\Phi'(\mathbf{v}) = \frac{1}{L_1^d} \sum_{i=1}^{k} \partial_i \phi((\mathbf{v}, \mathbf{g}_1)_H, \dots, (\mathbf{v}, \mathbf{g}_k)_H)\mathbf{g}_i \tag{2.2.13}$$

for the space-periodic cases. Since $\Phi'(\mathbf{v})$ is a linear combination of elements in V, $\Phi'(\mathbf{v})$ belongs to V.

Suppose now that the mapping

$$t \mapsto \int_V |\mathbf{v}|_V^2 \, d\mu_t(\mathbf{v}) \tag{2.2.14}$$

is integrable on the time interval $J = (0, T)$, i.e. that it belongs to $L^1(0, T)$. This implies that the family of measures $(\mu_t, t \in J)$ does not carry any mass on $H \backslash V$, i.e. $\mu_t(H \backslash V) = 0$ almost everywhere in t. Then it follows [2, chapt. 5.1] that the right-hand side of equation (2.2.10) is well defined and we may integrate equation (2.2.10) in time. This gives the following integral form

$$\int_H \Phi(\mathbf{v}) \, d\mu_t(\mathbf{v}) = \int_H \Phi(\mathbf{v}) \, d\mu_0(\mathbf{v}) + \int_0^t \int_H (\mathbf{F}(s, \mathbf{v}), \Phi'(\mathbf{v}))_H \, d\mu_s(\mathbf{v}) \, ds \tag{2.2.15}$$

for the no-slip case and

$$\int_H \Phi(\mathbf{v}) \, d\mu_t(\mathbf{v}) = \int_H \Phi(\mathbf{v}) \, d\mu_0(\mathbf{v}) + L_1^d \int_0^t \int_H (\mathbf{F}(s, \mathbf{v}), \Phi'(\mathbf{v}))_H \, d\mu_s(\mathbf{v}) \, ds \tag{2.2.16}$$

for the space periodic cases, and leads to the following energy-type inequality

$$\begin{aligned} \int_H |\mathbf{v}|_H^2 \, d\mu_t(\mathbf{v}) + 2\nu \int_0^t \int_V |\mathbf{v}|_V^2 \, d\mu_s(\mathbf{v}) \, ds \\ \leq \int_0^t \int_H (\mathbf{f}(s), \mathbf{v})_H \, d\mu_s(\mathbf{v}) \, ds + \int_H |\mathbf{v}|_H^2 \, d\mu_0(\mathbf{v}) \quad \text{for all } t \in [0, T]. \end{aligned} \tag{2.2.17}$$

We remark that for $d = 2$ we have equality in Equation (2.2.17) [2, chapt. 5.1].

2.3 Statistical solutions

The integral form (2.2.10), the energy-type inequality (2.2.17) and the fact that we should be able to calculate generalized moments from μ_t with any meaningful Φ lead to the following definition of statistical solutions of the Navier-Stokes equations (1.3.2) according to Foias-Prodi [1, def. 3.3][3, def. 3.2].

Definition 2.3.1. A one-parameter family $\mu = (\mu_t, t \in J)$ of Borel probability measures on H is called *statistical solution* of Equation (1.3.2) on $J \subset \mathbb{R}$ if

(i) the initial Borel probability measure μ_0 on H has *finite mean kinetic energy*, i.e.,

$$\int_H |\mathbf{v}|_H^2 \, d\mu_0(v) < \infty, \tag{2.3.1}$$

(ii) $\mathbf{f} \in L^2(J;H)$ and the Borel probability measures μ_t satisfy equation (2.2.10) for all $\Phi \in \mathcal{C}$ and the energy inequality (2.2.17) holds,

(iii) the mapping

$$J \ni t \mapsto \int_H \varphi(\mathbf{v})\, d\mu_t(\mathbf{v}) \tag{2.3.2}$$

is measurable on J for every bounded, continuous, real-valued function $\varphi : H \mapsto \mathbb{R}$ and the Borel probability measures $(\mu_t, t \in J)$ satisfy (compare [2, (V.1.12), (V.1.13)])

$$t \mapsto \int_V |\mathbf{v}|_V^2\, d\mu_t(\mathbf{v}) \in L^1(J), \quad t \mapsto \int_H |\mathbf{v}|_H^2\, d\mu_t(\mathbf{v}) \in L^\infty(J)\ . \tag{2.3.3}$$

(iv) (Liouville Equation) for every cylindrical testfunction Φ as in Definition 2.2.1, and for every $t, t' \in J$, μ_t satisfies

$$\begin{aligned}\int_H \Phi(\mathbf{v})\, d\mu_t(\mathbf{v}) = \int_H \Phi(\mathbf{v})\, d\mu_{t'}(\mathbf{v}) \\ + \int_{t'}^t \int_H (\mathbf{f}, \Phi'(\mathbf{v}))_H - \nu(A\mathbf{v}, \Phi'(\mathbf{v}))_H - (B(\mathbf{v},\mathbf{v}), \Phi'(\mathbf{v}))_H\, d\mu_s(\mathbf{v})\, ds.\end{aligned} \tag{2.3.4}$$

(v) (strengthened mean energy inequality (2.2.17)) on the time interval $J \subset \mathbb{R}$ there exists a subset $J' \subset J$ of full measure such that, for every nonnegative continuously differentiable function $\psi : [0,\infty) \to \mathbb{R}$ with $\|\psi'\|_{L^\infty((0,\infty))} < \infty$, there holds

$$\begin{aligned}\frac{1}{2}\int_H \psi(|\mathbf{u}|_H^2)\, d\mu_t(\mathbf{u}) + \nu \int_{t'}^t \int_H \psi'(|\mathbf{u}|_H^2)|\mathbf{u}|_V^2\, d\mu_s(\mathbf{u})\, ds \\ \le \frac{1}{2}\int_H \psi(|\mathbf{u}|_H^2)\, d\mu_{t'}(\mathbf{u}) + \int_{t'}^t \int_H \psi'(|\mathbf{u}(s)|_H^2)(\mathbf{f}(s), \mathbf{u}(s))_H\, d\mu_s(\mathbf{u})\, ds\end{aligned} \tag{2.3.5}$$

for every $t' \in J'$ and every $t \in J$ with $t' < t$.

The existence (and uniqueness in space dimension $d = 2$) of statistical solutions according to the previous definition is stated in the following result [2, thm. V.1.1-V.1.5].

Theorem 2.3.2. *Let μ_0 be a Borel probability measure on H with finite mean kinetic energy,*

$$\int_H |\mathbf{v}|_H^2\, d\mu_0(\mathbf{v}) < +\infty\ . \tag{2.3.6}$$

Let, moreover, $\mathbf{f} \in L^2(J;H)$ be a forcing term. Then, for either the no-slip case ($H = H_{dir}$) or the periodic case ($H = H_{per}$ or $H = \dot{H}_{per}$) there exists a statistical solution $(\mu_t, t \in J)$ of the Navier–Stokes equation on H in the sense of Definition 2.3.1.

In dimension $d = 2$, if μ_0 is supported in $B_H(R)$ for some $0 < R < \infty$, and if the forcing term $\mathbf{f} \in H$ is time-independent, the statistical solution is unique and explicitly given by $\mu_t = S(t,0)\mu_0$, for $t \ge t_0$.

Chapter 3

Monte Carlo method

We recall the important result from last chapter: assuming we are given a Borel probability measure μ_0 on H with finite mean kinetic energy and a forcing term $\mathbf{f} \in L^2(J;H)$, where $H = H_{dir}$, H_{per} or $\dot{H}_{per}$, then there exists a statistical solution $(\mu_t, t \in J)$ of the Navier–Stokes equations on H. We can get statistical information from the solution μ_t by using the generalized moments

$$\mathbb{E}_{\mu_t}(\Phi) = \int_H \Phi(\mathbf{v})\, d\mu_t(\mathbf{v}) \,, \quad t \in J \,, \tag{3.0.1}$$

where Φ is a bounded and continuous real-valued function on H.

Here we use the Monte Carlo method to approximate $\mathbb{E}_{\mu_t}(\Phi)$ numerically, as described in [1, sect. 4].

3.1 Monte Carlo method

We assume that we can sample from the exact initial distribution μ_0. We generate then $M \in \mathbb{N}$ independent copies $(\mathbf{v}^i, i = 1, \ldots, M)$ of $\mathbf{u}_0$, where $\mathbf{u}_0$ is μ_0-distributed. We assume further that for each sample $\mathbf{v}^i$, we can solve $\mathbf{v}(t) = S(t,0)\mathbf{v}^i$ exactly and that we can evaluate the real-valued functional $\Phi(\mathbf{v}(t))$ exactly. Then, we have the approximation

$$\mathbb{E}_{\mu_t}(\Phi) \approx E^M_{\mu_t}(\Phi) := \frac{1}{M} \sum_{i=1}^{M} \Phi(S(t,0)\mathbf{v}^i), \tag{3.1.1}$$

where we denote by $(E^M_{\mu_t}, M \in \mathbb{N})$ the sequence of Monte Carlo estimators which approximate the (generalized) expectation $\mathbb{E}_{\mu_t}(\Phi)$.

We assume that there is no forcing term, i.e. $\mathbf{f} = \mathbf{0}$. Also, we assume that Φ satisfies the *linear growth condition*, i.e., for some constant $C > 0$,

$$\forall \mathbf{v} \in H : \quad |\Phi(\mathbf{v})| \leq C(1 + |\mathbf{v}|_H) \,. \tag{3.1.2}$$

This is the case e.g. for all $\Phi \in \mathcal{C}$ (with $\mathcal{C}$ as in Definition 2.2.1) [1, sect. 4.1].

Then we have the following proposition, as stated and proved in [1, prop. 4.1].

Proposition 3.1.1. *Let $\Phi \in \mathcal{C}$ be a testfunction. Then, an error bound on the mean-square error of the Monte Carlo estimator $E^M_{\mu_t}$, for $M \in \mathbb{N}$, is given by*

$$\begin{aligned} \|\mathbb{E}_{\mu_t}(\Phi) - E^M_{\mu_t}(\Phi)\|_{L^2(H;\mathbb{R})} &= \frac{1}{\sqrt{M}} \left(Var_{\mu_t}(\Phi) \right)^{1/2} \\ &\leq C \frac{1}{\sqrt{M}} \Big(1 + \big(\int_H |v|_H^2 \, d\mu_0(v) \big)^{1/2} \Big). \end{aligned} \tag{3.1.3}$$

For $\nu > 0$, the latter inequality is strict.

We remark that the error estimate in Proposition 3.1.1 does not contain any implicit constant. It is therefore concluded in [1, sect. 4.1] that "the (mean-square over all flow configurations) *convergence rate* $[M^{-\frac{1}{2}}]$ *of Monte Carlo sample averages is uniform with respect to the physical parameters of the flow* [e.g. ν] but depends, of course, on the second moment of μ_0, i.e. on the mean kinetic energy of the initial probability measure μ_0".

The error bound in Proposition 3.1.1 is semi-discrete, in the sense that it requires an exact (weak) solution of the Navier–Stokes equations for each initial velocity sample drawn from μ_0. But in order to obtain "computationally feasible approximations" of generalized moments of statistical solutions, we have to perform additional space and time discretizations [1, sect. 5.0]. This adds a bias to the error bound in Proposition 3.1.1, as we shall see in the next chapter.

To conclude this chapter, we discuss the issue of sampling exactly from μ_0, since the calculation of the Monte Carlo estimator was based on this assumption. As mentioned in [1, sect. 4.1], this is "not a constraint" if μ_0 is given by a finite-dimensional measure. But we also have to be able to sample from a measure μ_0 defined on a possibly infinite-dimensional space.

3.2 Discretization of the initial distribution μ_0

We recall that the initial distribution μ_0 is defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and is assumed to be given as an image measure under an H-valued random variable $\mathbf{X}$ with distribution μ_0, where the random variable $\mathbf{X}$ is defined as a mapping from the measurable space $(\Omega, \mathcal{F})$ into the measurable space $(H, \mathcal{B}(H))$ such that $\mu_0 = \mathbf{X} \circ \mathbb{P}$ [1, sect. 7.0].

We assume for simplicity that μ_0 is a Gaussian measure supported on H or on a subspace of H. Since Gaussian measures are completely characterized by the mean $\mathbf{m} \in H$ and the covariance operator Q defined on H [1, sect. 7.0], the Gaussian random variable $\mathbf{X}$ is given by the Karhunen-Loève expansion

$$\mathbf{X} = \mathbf{m} + \sum_{i \in \mathbb{N}} \sqrt{\lambda_i} \beta_i \mathbf{w}_i, \tag{3.2.1}$$

where $((\lambda_i, \mathbf{w}_i), i \in \mathbb{N})$ is a complete orthonormal system in H and consists of eigenvalues and eigenfunctions of Q, and the sequence $(\beta_i, i \in \mathbb{N})$ consists of real-valued, independent, standard normal-distributed random variables [1, sect. 7.0].

Because the expansion in (3.2.1) is infinite, in order to generate $\mathbf{X}$ numerically, we use a truncated expansion of the form

$$\mathbf{X}^\kappa = \mathbf{m} + \sum_{i=1}^{\kappa} \sqrt{\lambda_i} \beta_i \mathbf{w}_i \, , \tag{3.2.2}$$

with $\kappa \in \mathbb{N}$, mean $\mathbf{m} \in H$ and covariance operator Q^κ. The sequence of truncated sums $(\mathbf{X}^\kappa, \kappa \in \mathbb{N})$ converges $\mathbb{P}$-a.s. to $\mathbf{X}$ for $\kappa \to +\infty$ [1, sect. 7.0], and the $L^2(\Omega; H)$-error of this truncation is controlled by the decay of the eigenvalues, as shown in the following lemma, stated and proved in [1, lemm. 7.1].

Lemma 3.2.1. *If the eigenvalues $(\lambda_i, i \in \mathbb{N})$ of the covariance operator Q of the Gaussian random variable $\mathbf{X}$ on H have a rate of decay of $\lambda_i \leq C\, i^{-\gamma}$, then the sequence $(\mathbf{X}^\kappa, \kappa \in \mathbb{N})$ converges to $\mathbf{X}$ in $L^2(\Omega; H)$ and the error is bounded by*

$$\|\mathbf{X} - \mathbf{X}^\kappa\|_{L^2(\Omega;H)} \leq C\, \frac{1}{\sqrt{\gamma - 1}} \kappa^{-\frac{\gamma-1}{2}}. \tag{3.2.3}$$

3.2.1 Expansion in terms of Stokes eigenfunctions

We consider now the space-periodic case with vanishing space average, with $H = \dot{H}_{per}$, and let A be the corresponding Stokes operator (see Chapter 1, Section 1.4.4). We recall that we can define fractional powers A^a of A, with $a \in \mathbb{R}$, and that the eigenfunctions of A form an orthonormal basis of H.

By choosing $Q = A^{-\delta}$ and prescribing a mean velocity field $\langle \mathbf{u}_0 \rangle = \mathbb{E}_{\mu_0}(H) \in H$, draws of the random initial velocity $\mathbf{u}_0$ with law μ_0 can be obtained from the Karhunen-Loève expansion (see (3.2.1))

$$\mathbf{u}_0(\omega; \mathbf{x}) = \langle \mathbf{u}_0 \rangle + \sum_{i \in \mathbb{N}} \sqrt{\mu_i} \xi_i(\omega) \mathbf{w}_i(\mathbf{x}) \,, \tag{3.2.4}$$

where $\mathbf{w}_i \in V$ denote the eigenfunctions of the Stokes operator A, $\xi_i \sim \mathcal{N}(0,1)$ are independent standard normal random variables taking values in $\mathbb{R}$, and μ_i are the Karhunen-Loève eigenvalues, which, by the spectral mapping theorem, are given by $\mu_i = \lambda_i^{-\delta}$, where the $(\lambda_i, i \in \mathbb{N})$ are the eigenvalues of the Stokes operator A [1, sect. 7.2].

In our numerical experiments, we will obtain draws of the random initial velocities $\mathbf{u}_0$ by using a truncated expansion of the form

$$\mathbf{u}_0(\omega; \mathbf{x}) = \sum_{i=1}^{\kappa} \sqrt{\lambda_i} Y_i(\omega) \mathbf{w}_i(\mathbf{x}) \,, \tag{3.2.5}$$

where $\kappa < \infty$ and Y_i are independent and *uniformly* distributed random variables on a bounded interval (a, b). The expectation of $\mathbf{u}_0$ is then

$$\mathbb{E}(\mathbf{u}_0) = \sum_{i=1}^{\kappa} \sqrt{\lambda_i} \mathbb{E}(\beta_i) \mathbf{w}_i = \frac{a+b}{2} \sum_{i=1}^{\kappa} \sqrt{\lambda_i} \mathbf{w}_i \,, \tag{3.2.6}$$

and its (squared) $L^2(\Omega; H)$-norm is

$$\begin{aligned} \| \sum_{i=1}^{\kappa} \sqrt{\lambda_i} Y_i \mathbf{w}_i \|^2_{L^2(\Omega;H)} &= \mathbb{E}\big(\| \sum_{i=1}^{\kappa} \sqrt{\lambda_i} Y_i \mathbf{w}_i \|^2_H \big) \\ &= \sum_{i=1}^{\kappa} \lambda_i \mathbb{E}(Y_i^2) \|\mathbf{w}_i\|^2_H \\ &= \frac{a^2 + ab + b^2}{3} \sum_{i=1}^{\kappa} \lambda_i \,. \end{aligned} \tag{3.2.7}$$

We choose $\lambda_i = C i^{-\gamma}$, with $\gamma > 1$ such that $\mathbf{u}_0$ is still in $L^2(\Omega; H)$ (resp. the initial probability distribution has finite kinetic energy) when $\kappa \to \infty$. The truncation error with respect to the case $\kappa = \infty$ is bounded the same way as in Lemma 3.2.1.

The reason we choose uniformly distributed coefficients Y_i instead of normally distributed ones is that by sampling from a normal distribution, we may obtain large samples Y_i and thus large values for the initial velocities. This leads to high Reynolds number flows that may be difficult for the discrete solvers to handle, or could lead to compatibility issues (in the sense that the velocities in the initial ensemble should satisfy Ma $\lesssim 0.3$ for the incompressible Navier-Stokes equations to be applicable (see Chapter 1, Section 1.3)) or even to an unphysical problem (velocities higher than the speed of light).

Chapter 4

Space and time discretization

In Chapter 3, Section 3.1, for the computation of the Monte Carlo estimator for the generalized moment $\mathbb{E}_{\mu_t}(\Phi)$ at time t, we assumed that we could calculate exactly the solution $S(t,0)\mathbf{v}^i$ from a random initial velocity $\mathbf{v}^i$ drawn from the probability distribution μ_0.

In this chapter, we follow the lines of [1, sect. 5-6], and address the effect of space and time discretizations, used to compute the pathwise solutions numerically, on the mean-square error of the Monte Carlo (MC) estimator. Then we present the multilevel Monte Carlo (MLMC) method, which uses a hierarchic family of discretizations in space and time to equilibrate statistical and discretization errors more efficiently than the Monte Carlo method.

4.1 Fully-discrete formulation

We summarize here the description presented in [1, sect. 5.1-2].

For the discretization in space, a nested family of finite dimensional subspaces $\mathcal{V} = (V_\ell, \ell \in \mathbb{N}_0)$ of $L^2(D)$ is introduced. The subspaces V_ℓ are endowed with the canonical inner product of $L^2(D)$, which is also the H-norm (see Chapter 1, Section 1.4.1). The refinement levels $\ell \in \mathbb{N}_0$, the refinement sizes $(h_\ell, \ell \in \mathbb{N}_0)$ and the projections $(P_\ell, \ell \in \mathbb{N}_0)$ from V onto V_ℓ are associated to the subspaces V_ℓ. For $\ell \in \mathbb{N}_0$, the sequence is supposed to be dense in the sense that

$$\lim_{\ell \to +\infty} |\mathbf{v} - P_\ell \mathbf{v}_\ell|_H = 0 \quad \forall \mathbf{v} \in V \ . \tag{4.1.1}$$

For the discretization in time, a sequence of time discretizations $\Theta = (\Theta_\ell, \ell \in \mathbb{N}_0)$ of the time interval $[0,T]$, for $T < +\infty$, is introduced, each of equidistant or maximum time steps of size $\Delta_\ell t$. The time discretization at level $\ell \in \mathbb{N}_0$, Θ_ℓ, is the partition of $[0,T]$ which is given by

$$\Theta_\ell = \{t_\ell^i \in [0,T] : \ t_\ell^i = i \cdot \Delta_\ell t, \ i = 0, \ldots, \frac{T}{\Delta_\ell t}\} \ . \tag{4.1.2}$$

We denote by $S_\ell = (S_\ell(t_\ell^i, 0), \ i = 0, \ldots, T/\Delta_\ell t)$ the full-discrete solution operator that maps $\mathbf{u}_0$ into $\mathbf{u}_{\ell,\ell} = (\mathbf{u}_{\ell,\ell}(t_\ell^i), \ i = 0, \ldots, T/\Delta_\ell t)$. The spaces in $\mathcal{V}$ and the time discretizations Θ are assumed to be chosen such that the following error bound holds.

Assumption 4.1.1. The sequence of full-discrete solutions $(\mathbf{u}_{\ell,\ell}, \ell \in \mathbb{N}_0)$ converges to the (unique, in space dimension $d = 2$) solution $\mathbf{u}$ of Equation (1.3.2). The space and time discretization error is bounded, for $\ell \in \mathbb{N}$ and $t \in \Theta_\ell$, by

$$|\mathbf{u}(t) - \mathbf{u}_{\ell,\ell}|_H = |S(t,0)\mathbf{u}_0 - S_\ell(t,0)\mathbf{u}_0|_H \leq C\,(\frac{h_\ell^\sigma}{\nu} + \frac{(\Delta_\ell t)^\sigma}{\nu}), \tag{4.1.3}$$

for $d \geq 2$ and for $\sigma > 0$.

In both cases $C > 0$ is independent of ν, ℓ and h_ℓ. With the choice $h_\ell \simeq \Delta_\ell t$ this reduces to

$$|\mathbf{u}(t) - \mathbf{u}_{\ell,\ell}|_H \leq C \, \frac{h_\ell^\sigma}{\nu}, \tag{4.1.4}$$

for $d \geq 2$ and for $\sigma > 0$.

Convergence requirement

Achieving asymptotic convergence in Assumption 4.1.1 requires that there exists $\ell^* \in \mathbb{N}_0$ such that $h_{\ell^*}^\sigma \leq \nu$. Then for all h_ℓ with $\ell \geq \ell^*$, we say that the convergence requirement is fulfilled and that the refinement levels $\ell \geq \ell^*$ are resolved. If we are in the regime $\ell < \ell^*$ (which implies $h_\ell^\sigma > \nu$), the convergence requirement is not fulfilled, and we say that the refinement levels $\ell < \ell^*$ are under-resolved.

Remark 4.1.2. The assumption of a space and time discretization with the convergence bound (4.1.4) where the rate of convergence $\sigma > 0$ holds for large Reynolds numbers (robust convergence [1, sect. 5.1]) and where the constant $C > 0$ is independent of the fluid viscosity in the norm $L^\infty(J; H)$ is "strong" [1, rem. 5.3]. It means essentially that the numerical scheme resolves the bulk properties of the flow consistent to order $\sigma > 0$ "independent of the small scale features of the flow" [1, rem. 5.3]. In practice, therefore, Assumption 4.1.1 implies that, "for flows with large Reynolds number, a proper turbulence model is used for discretizations which do not resolve physical length scales of the flow" [1, rem. 5.3].

4.1.1 Discretization with Finite Differences

To compute numerically the pathwise solutions, we use in this thesis a solver named 'IMPACT'. It is a massively parallel solver for incompressible flows which uses Finite Differences in both space and time for the discretization. We refer to Chapter 5 for a detailed description of this solver.

4.2 Multilevel Monte Carlo method

4.2.1 Singlelevel Monte Carlo method

We recall that on each discretization level $\ell \in \mathbb{N}_0$, we have a space and time discretization V_ℓ and Θ_ℓ and a corresponding discrete solution operator for computing the discrete pathwise solutions $S_\ell(t, 0)\mathbf{u}_0 = \mathbf{u}_{\ell,\ell} \in V_\ell$, with $t \in \Theta_\ell$. We can then formulate the fully discrete Monte Carlo estimator on level ℓ with M_ℓ samples

$$E_{\mu_t}^{M_\ell}(\Phi) \approx E_{\mu_t}^{M_\ell}(\Phi_\ell) := \frac{1}{M_\ell} \sum_{i=1}^{M_\ell} \Phi(S_\ell(t, 0)\mathbf{v}^i). \tag{4.2.1}$$

This approach is called singlelevel Monte Carlo [1, sect. 6.1], since all samples of the Monte Carlo estimator are approximated with one common space and time discretization. As we shall see next, the space and time discretization introduces a bias in the mean square error bound of the (discrete) Monte Carlo estimator.

We assume here that the testfunction $\Phi \in \mathcal{C}$ satisfies a *Lipschitz condition*, i.e. there exists $C > 0$ such that

$$\forall \mathbf{u}, \mathbf{v} \in H : \quad |\Phi(\mathbf{u}) - \Phi(\mathbf{v})| \leq C|\mathbf{u} - \mathbf{v}|_H. \tag{4.2.2}$$

We remark that Equation (4.2.2) is an additional constraint compared to the linear growth condition in Equation (3.1.2).

The Monte Carlo estimator has then the following mean-square error bound, as stated and proved in [1, thm. 6.1].

Theorem 4.2.1. *If, for $\Phi \in \mathcal{C}$ fulfilling Equation (4.2.2) and $\ell \in \mathbb{N}_0$, the fully-discrete Monte Carlo estimator $E^{M_\ell}_{\mu_t}(\Phi_\ell)$ for the generalized moment of the statistical solution fulfills Assumption 4.1.1, for $\sigma > 0$ and $h_\ell \simeq \Delta_\ell t$, then the variance of the estimator admits, for $t \in \Theta_\ell$, the bound*

$$\begin{aligned} \|\mathbb{E}_{\mu_t}(\Phi) - E^{M_\ell}_{\mu_t}(\Phi_\ell)\|_{L^2(H;\mathbb{R})} &\leq \frac{1}{\sqrt{M_\ell}}\big(Var_{\mu_t}(\Phi)\big)^{1/2} + \|\Phi - \Phi_\ell\|_{L^2(H;\mathbb{R})} \\ &\leq C\,\big(\frac{1}{\sqrt{M_\ell}} + \frac{h_\ell^\sigma}{\nu}\big). \end{aligned} \tag{4.2.3}$$

The constant $C > 0$ is independent of ℓ, h_ℓ and of ν.

We see in Theorem 4.2.1 that the error bound for the singlelevel Monte Carlo estimator consists of two additive components, one for the 'sampling' error (which is the same as in Proposition 3.1.1) and one for the 'discretization' error. Although only an upper bound is stated, this error is indeed of additive nature [1, sect. 6.1]. That is, in order to achieve that the total error in Theorem 4.2.1 is smaller than a prescribed tolerance $\epsilon > 0$, we require that, for some $\eta \in (0, 1)$,

$$\frac{1}{\sqrt{M_\ell}}\big(\mathrm{Var}_{\mu_t}(\Phi)\big)^{1/2} \leq \eta \cdot \epsilon \;\text{ and }\; \|\Phi - \Phi_\ell\|_{L^2(H;\mathbb{R})} \leq (1-\eta)\epsilon\,. \tag{4.2.4}$$

For example, in order equilibrate statistical and discretization errors and achieve an error of the order of magnitude of the discretization error, we set the number of samples M_ℓ to

$$M_\ell = O\big((\frac{\nu}{h_\ell^\sigma})^2\big), \tag{4.2.5}$$

where all constants implied in the Landau symbol $O(\cdot)$ are independent of ν, ℓ and h_ℓ [1, sect. 6.2].

4.2.2 Multilevel Monte Carlo method

In the singlelevel approach, we calculated all samples of the Monte Carlo estimator on a single level of discretization in space and time. We extend now this approach to a multilevel discretization, where samples of the Monte Carlo estimator are drawn and calculated on a hierarchy of nested spatial and temporal discretizations [17, sect. 1].

The idea is that the expectation of the discrete solution Φ_L on some discretization level L, for $t \in \Theta_L$, can be written (telescopic sum) as

$$\mathbb{E}_{\mu_t}(\Phi_\ell) = \mathbb{E}_{\mu_t}(\Phi_0) + \sum_{\ell=1}^{L} \mathbb{E}_{\mu_t}(\Phi_\ell - \Phi_{\ell-1}). \tag{4.2.6}$$

That is, it can be expanded as the expectation on the (coarsest) discretization level $\ell = 0$ and a sum of correcting terms on all discretization levels $\ell = 1, \ldots, L$.

The expectation of each term on the right-hand side is then approximated with a Monte Carlo estimator, with a corresponding level-dependent number of samples M_ℓ

$$\begin{aligned} E^L_{\mu_t}(\Phi_L) &= E^{M_0}_{\mu_t}(\Phi_0) + \sum_{\ell=1}^{L} E^{M_\ell}_{\mu_t}(\Phi_\ell - \Phi_{\ell-1}) \\ &= \frac{1}{M_0}\sum_{i=1}^{M_0} \Phi(S_0(t,0)\mathbf{v}^i) + \sum_{\ell=1}^{L}\frac{1}{M_\ell}\sum_{i=1}^{M_\ell}(\Phi(S_\ell(t,0)\mathbf{v}^i) - \Phi(S_{\ell-1}(t,0)\mathbf{v}^i))\,. \end{aligned} \tag{4.2.7}$$

The term $E^L_{\mu_t}$ is called the multilevel Monte Carlo estimator for discretization level $L \in \mathbb{N}_0$ [1, sect. 6.2]. We remark that for the difference $\Phi(S_\ell(t,0)\mathbf{v}^i) - \Phi(S_{\ell-1}(t,0)\mathbf{v}^i)$ appearing on the right-hand side, for a given i, the *same* initial velocity sample $\mathbf{v}^i$ is taken.

The multilevel Monte Carlo estimator has the following mean-square error bound, as stated and proved in [1, prop. 6.3].

Proposition 4.2.2. *If, for $\Phi \in \mathcal{C}$ fulfilling Equation (4.2.2) and $L \in \mathbb{N}_0$, the fully-discrete Monte Carlo estimator $E^{M_\ell}_{\mu_t}(\Phi_\ell)$ for the generalized moment of the statistical solution fulfills Assumption 4.1.1, for all $\ell = 0, \dots, L$ with $\sigma > 0$ and $h_\ell \simeq \Delta_\ell t$, and if for $\ell = 0, \dots, L$ $h_{\ell-1} \leq \varrho h_\ell$, with some reduction factor $0 < \varrho < 1$ independent of ℓ, then there exists $C(\varrho) > 0$ independent of L, such that the the variance of the estimator admits, for $t \in \Theta_L$, the error bound*

$$\begin{aligned}\|\mathbb{E}_{\mu_t}(\Phi) - E^L_{\mu_t}(\Phi_L)\|_{L^2(H;\mathbb{R})} &\leq \|\Phi - \Phi_L\|_{L^2(H;\mathbb{R})} + \sum_{\ell=0}^{L} \frac{1}{\sqrt{M_\ell}} \big(Var_{\mu_t}(\Phi_\ell - \Phi_{\ell-1}) \big)^{1/2} \\ &\leq C(\varrho) \Big(\frac{h_L^\sigma}{\nu} + \frac{1}{\sqrt{M_0}} + \sum_{\ell=0}^{L} \frac{1}{\sqrt{M_\ell}} \frac{h_\ell^\sigma}{\nu} \Big).\end{aligned} \tag{4.2.8}$$

In order to achieve that the total error in Proposition 4.2.2 is smaller than a prescribed tolerance $\epsilon > 0$, we require that, for some $\eta_L \in (0,1)$,

$$\|\Phi - \Phi_L\|_{L^2(H;\mathbb{R})} \leq (1 - \eta_L)\, \epsilon, \tag{4.2.9}$$

and

$$\sum_{\ell=0}^{L} \frac{1}{\sqrt{M_\ell}} \big(\mathrm{Var}_{\mu_t}(\Phi_\ell - \Phi_{\ell-1}) \big)^{1/2} \leq \eta_L\, \epsilon. \tag{4.2.10}$$

We determine next the required number M_ℓ of Monte Carlo samples on each discretization level ℓ, in order to equilibrate the errors arising from each term $\mathrm{Var}_{\mu_t}(\Phi_\ell - \Phi_{\ell-1})$ and achieve a total error of the order of the discretization error.

We suppose that the convergence requirement is at least fulfilled on the finest level, i.e., $h_L^\sigma < \nu$. Then, on the first level,

$$M_0 = O\left(\Big(\frac{\nu}{h_L^\sigma} \Big)^2 \right) \tag{4.2.11}$$

is chosen, in order to equilibrate the statistical and the discretization error contributions [1, sect. 6.2]. The sample numbers M_ℓ, for the discretization levels $\ell = 1, \dots, L$, is chosen according to

$$M_\ell = O\left(\Big(\frac{h_\ell^\sigma}{h_L^\sigma} \Big)^2 \ell^{2(1+\eta)} \right), \tag{4.2.12}$$

for $\eta > 0$ [1, sect. 6.2].
All constants implied in the Landau symbols $O(\cdot)$ are independent of ν [1, sect. 6.2].

Cost considerations

On each level $\ell = 1, \dots, L$, the cost $\mathcal{W}_\ell$ to compute $\sum_{\ell=1}^{L} E^{M_\ell}_{\mu_t}(\Phi_\ell - \Phi_{\ell-1})$ is M_ℓ times the (average) cost to calculate one discrete solution $\Phi(S_\ell(t,0)\mathbf{v})$ and one discrete solution $\Phi(S_{\ell-1}(t,0)\mathbf{v})$ on the discretization level ℓ. And on level $\ell = 0$, the cost $\mathcal{W}_0$ is M_0 times the (average) cost to calculate one discrete solution $\Phi(S_0(t,0)\mathbf{v})$ on the coarsest discretization

level. If the costs W_ℓ, $\ell = 0, \ldots, L-1$, are smaller than or equal to the cost W_L on the finest discretization level, then, with a small number of samples M_L, the cost for calculating the multilevel Monte Carlo estimator $E^L_{\mu_t}(\Phi_L)$ is approximatively only a couple of times the cost to calculate one discrete solution on the finest discretization level L.

In comparison, in the singlelevel Monte Carlo approach, we need to calculate usually more (see Equation (4.2.5)) fine discrete solutions in order to equilibrate statistical and discretization error. With a Finite Volume solver used to calculate the discrete solutions, it was shown that the multilevel Monte Carlo approach was, at the relative error level of 1%, "two orders of magnitude faster" than the singlelevel approach ([17, sect. 1] and references there).

In this thesis, we compute numerically the discrete pathwise solutions $S_\ell(t, 0)\mathbf{v}$ with a massively parallel solver for incompressible flows named 'IMPACT', which uses Finite Differences in both space and time for the discretization and solves resulting linear systems of equations iteratively. We describe this solver more in detail in the next chapter.

Chapter 5

IMPACT

The IMPACT code was originally developed by Dr. Rolf Henniger at ETH Zurich in the context of his Phd thesis "Direct and Large-Eddy Simulation of particle transport processes in estuarine environments" [4] in the research group of Prof. L. Kleiser. The code is continuously developed at the Institute of Fluid Dynamics at ETH Zurich [5].

Utilization of the code in the context of this thesis was kindly permitted by Prof. L. Kleiser and PD Dr. D. Obrist.

This chapter aims at summarizing the code capabilities that may be of interest in the context of this thesis as well as the underlying theoretical and algorithmic concepts, as they were described in [4] and [6], and follows closely the descriptions, notations, equations and figures found there.

5.1 General description

IMPACT stands for a simulation code that can predict the evolution of "**I**ncompressible (turbulent) flows by means of **M**assively **PA**rallel **C**ompu**T**ers" [4, chapt. 2.0].

5.1.1 Governing equations

The code solves the Navier-Stokes equations for incompressible flows in dimension $d = 2$ or 3 given by

$$\frac{\partial}{\partial t}\mathbf{u} = -\nabla p + \underbrace{\frac{1}{\mathrm{Re}}\triangle\mathbf{u}}_{\mathcal{L}\mathbf{u}} + \underbrace{\mathbf{f} - (\mathbf{u}\cdot\nabla)\mathbf{u}}_{\mathcal{N}(\mathbf{u})} \tag{5.1.1a}$$

$$\nabla\cdot\mathbf{u} = 0\ , \tag{5.1.1b}$$

for an initial condition $\mathbf{u}_0$ and appropriate boundary conditions for $\mathbf{u}$.

The momentum and continuity equations (5.1.1a) and (5.1.1b) can be written in matrix form (including boundary conditions), by

$$\frac{\partial}{\partial t}\begin{bmatrix}\mathbf{u}\\0\end{bmatrix} = \begin{bmatrix}\mathcal{L} & -\mathcal{G}\\ -\mathcal{D} & 0\end{bmatrix}\begin{bmatrix}\mathbf{u}\\p\end{bmatrix} + \begin{bmatrix}\mathcal{N}(\mathbf{u})\\0\end{bmatrix}\ , \tag{5.1.2}$$

where $\mathcal{D}$ and $\mathcal{G}$ are resp. the divergence and gradient operators.

By applying the continuity equation (5.1.1b) to the momentum equation (5.1.1a) we obtain an equation for the pressure:

$$\triangle p = \nabla\cdot\mathcal{N}(\mathbf{u})\ . \tag{5.1.3}$$

The code solves a discretized form of (5.1.2), and from this discretized system of equations, a discretized equation for the pressure is derived such that no explicit pressure boundary conditions need to be specified [6, sect. 2.0].

The governing equations are solved on a rectangular domain $D = (0, L_1) \times (0, L_2) \times (0, L_3)$ with boundary ∂D and extents L_1, L_2 and L_3. Suitable boundary conditions for the velocity $\mathbf{u}$ that are handled by the code are:

- Periodic boundary conditions.
- Symmetry boundary conditions.
- Dirichlet boundary conditions $\mathbf{u}(\mathbf{x}, t)|_{\partial D} = \mathbf{g}(\mathbf{x}, t)$.
  Note that the following *compatibility condition* must be satisfied [4, chapt. 2.1]:

$$\int_{\partial D} \mathbf{g} \cdot \mathbf{n} \, dS = 0 \, . \tag{5.1.4}$$

- Advective boundary conditions $\frac{\partial}{\partial t}\mathbf{u}(\mathbf{x}, t) + c(\mathbf{x}) \frac{\partial}{\partial \mathbf{n}}\mathbf{u}(\mathbf{x}, t)\big|_{\partial D} = -\mathcal{A}(\mathbf{x}, t)$.

As previously mentioned, the boundary conditions for the pressure p in Equation (5.1.3) depend implicitly on the aforementioned boundary conditions for the velocity $\mathbf{u}$.

As mentioned in [6, sect. 3.0]: "A complete strategy for solving numerically the incompressible Navier-Stokes equations consists of a data decomposition method, a discretization scheme and an appropriate solution technique for the resulting system of linear equations". We present next the domain decomposition method.

5.2 Domain decomposition and datastructure

IMPACT uses a *static* data decomposition as sketched in Figure 5.1 for a 2D problem [4, chapt. 2.2.1].

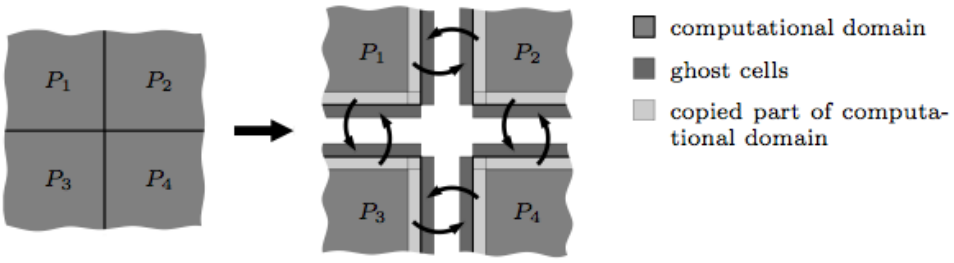


Figure 5.1: Static data decomposition and ghost cell update between four processors. Figure and caption taken from [4, fig. 2.1].

The computational domain is decomposed into sub-domains on a cartesian grid (see Figure 5.1) and each processor is mapped to one of the sub-domains and holds it in its memory. The connection with sub-domains is done by ghost cells which are located at the junctions between the sub-domains. Each sub-domain contains only a portion of the discrete global vectors and operators (e.g. the diagonal blocks in a system of linear equations), and the ghost cells correspond to the parts of the operator which cannot be distributed (e.g. the off-diagonal blocks in a system of linear equations) [4, chapt. 2.2.1]. Before a global operator is applied to a global vector, the data in the ghost cells is updated or synchronized with the corresponding data from the neighboring processors [4, chapt. 2.2.1].

5.3 Temporal discretization scheme

5.3.1 Stability and efficiency

The maximum time step size for a stable time integration of Equation (5.1.2) with an explicit time integration scheme is estimated from the Courant-Friedrichs-Lévy (CFL) condition [6, sect. 4.1]. If only the convective term $(\mathbf{u}\cdot\nabla)\mathbf{u}$ is taken into account, for e.g. in the semi-implicit scheme used in the code [4, chapt. 2.3.1], the convective time step limit (for $d=3$, similarly for $d=2$) is set [6, sect. 4.1] to

$$\begin{aligned} \Delta t &\leq \frac{s_{\text{conv}}}{\max\limits_{D}\left\{|u_1|\tilde{\kappa}_{C,1}+|u_2|\tilde{\kappa}_{C,2}+|u_3|\tilde{\kappa}_{C,3}\right\}} \\ \Leftrightarrow \quad \Delta t &= \tilde{\text{CFL}}\frac{s_{\text{conv}}}{\max\limits_{D}\left\{|u_1|\tilde{\kappa}_{C,1}+|u_2|\tilde{\kappa}_{C,2}+|u_3|\tilde{\kappa}_{C,3}\right\}} , \end{aligned} \tag{5.3.1}$$

with the 'normalized' CFL–number $0<\tilde{\text{CFL}}\leq 1$. The parameter s_{conv} is the stability limit of the time integration scheme used to treat the convective term, which is known or can be precalculated before the computations are started. The $\tilde{\kappa}_{C,i}=\tilde{\kappa}_{C,i}(\Delta x)$, for $i=1,2,3$, are the maximum modified wave numbers of the spatial discretization of the convective term, which are calculated before the start of the time integration. For our simulations, we set the value of $\tilde{\text{CFL}}$ to 0.75.

We remark that Equation (5.3.1) leads also to a more 'classical' CFL-condition [6, sect. 4.1], namely

$$\Delta t \leq \text{CFL}_{\text{conv}}\ \frac{1}{\max\limits_{D}\left\{\frac{|u_1|}{\Delta x_1}+\frac{|u_2|}{\Delta x_2}+\frac{|u_3|}{\Delta x_3}\right\}} , \tag{5.3.2}$$

where the CFL–number

$$\text{CFL}_{\text{conv}} \equiv \frac{s_{\text{conv}}}{\max\limits_{D}\left\{|\Delta x_1|\tilde{\kappa}_{C,1}+|\Delta x_2|\tilde{\kappa}_{C,2}+|\Delta x_3|\tilde{\kappa}_{C,3}\right\}} \tag{5.3.3}$$

is calculated (or a lower bound is guessed) before the start of the time integration.

If only the diffusive term $\triangle\mathbf{u}$ is taken into account, the viscous time step limit (for $d=3$, similarly for $d=2$) is set [6, sect. 4.1] to

$$\Delta t \leq \frac{s_{\text{visc}}}{\max\limits_{D}\left\{\frac{1}{\text{Re}}(\tilde{\kappa}_{L,1}^2+\tilde{\kappa}_{L,2}^2+\tilde{\kappa}_{L,3}^2)\right\}} . \tag{5.3.4}$$

Here, the parameter s_{visc} is the stability limit of the time integration scheme used to treat the viscous term, which is known or can be precalculated before the computations are started. The $\tilde{\kappa}_{L,i}=\tilde{\kappa}_{C,i}(\Delta x)$, for $i=1,2,3$, are the maximum modified wave numbers of the spatial discretization of the viscous term, which are calculated before the start of the time integration.

We remark that Equation (5.3.4) leads also to a more 'classical' CFL-condition [6, sect. 4.1], given by

$$\Delta t \leq \text{CFL}_{\text{visc}}\ \frac{1}{\max\limits_{D}\left\{\frac{1}{\text{Re}}\left(\frac{1}{\Delta x_1^2}+\frac{1}{\Delta x_2^2}+\frac{1}{\Delta x_3^2}\right)\right\}} , \tag{5.3.5}$$

where the CFL–number

$$\text{CFL}_{\text{visc}} \equiv \frac{s_{\text{visc}}}{\max\limits_{D}\left\{(\Delta x_1\tilde{\kappa}_{L,1})^2+(\Delta x_2\tilde{\kappa}_{L,2})^2+(\Delta x_3\tilde{\kappa}_{L,3})^2\right\}} \tag{5.3.6}$$

is again calculated (or a lower bound is guessed) before the start of the time integration.

If both the convective and diffusive terms are taken into account, for e.g. in the fully explicit scheme used in the code [4, chapt. 2.3.1], the time step limit (for $d = 3$, similarly for $d = 2$) is set [4, chapt. 2.3.1] to

$$\Delta t = \tilde{\mathrm{CFL}} \frac{s_{\mathrm{conv+visc}}}{\max\limits_{D} \Big\{ \sqrt{\Big| |u_1|\tilde{\kappa}_{C,1} + |u_2|\tilde{\kappa}_{C,2} + |u_3|\tilde{\kappa}_{C,3} \Big|^2 + \Big| \frac{1}{\mathrm{Re}} (\tilde{\kappa}^2_{L,1} + \tilde{\kappa}^2_{L,2} + \tilde{\kappa}^2_{L,3}) \Big|^2 } \Big\}} \ , \tag{5.3.7}$$

with the 'normalized' CFL–number $0 < \tilde{\mathrm{CFL}} \leq 1$. The parameter $s_{\mathrm{conv+visc}}$ is the stability limit of the time integration scheme used to treat both convective and diffusive terms, which is known or can be precalculated before the computations are started. For our simulations, we set the value of $\tilde{\mathrm{CFL}}$ to 0.75.

As it can be seen from the time step limits (5.3.1) and (5.3.4), there is "always a Reynolds number Re (and an according fine-grid spacing Δx) below which the viscous time step limit is more restrictive than the convective limit" [6, sect. 4.1]. Such viscous time step size restrictions can be avoided by using an implicit time integration scheme [4, chapt. 2.3.1]. This results in solving an additional linear system of equations, which increases the computational work per time step [6, sect. 4.1]. However, the time step sizes may be larger than with an explicit time integration (due to the less restrictive stability limit), such that less time steps are needed to advance the solution over a given time interval. As mentioned in [4, chapt. 2.3.1], it is "often hard to judge beforehand whether implicit or explicit time integration is more efficient overall" since "accuracy requirements may impose stronger limitations on the time step size than the stability limits".

5.3.2 Integration scheme

We recall here that the momentum and continuity equations (5.1.1a) and (5.1.1b) can be written in matrix form (including boundary conditions):

$$\frac{\partial}{\partial t} \begin{bmatrix} \mathbf{u} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathcal{L} & -\mathcal{G} \\ -\mathcal{D} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} + \begin{bmatrix} \mathcal{N}(\mathbf{u}) \\ 0 \end{bmatrix} \ , \tag{5.3.8}$$

where $\mathcal{D}$ and $\mathcal{G}$ are resp. the divergence and gradient operators.

Let $\mathbf{u}^{(0)} = \mathbf{u}(t)$ be the solution at time t. For advancing the solution by a time step size Δt, the (CN-)RK3 ((Crank-Nicolson)-Runge-Kutta 3) scheme is used for the system (5.3.8) [4, chapt. 2.3.1], and it reads:

$$\begin{aligned} \frac{1}{\Delta t}(\mathbf{u}^{(m)} - \mathbf{u}^{(m-1)}) = \alpha_c^{(m)} [\Theta_{CN} \mathcal{L}\mathbf{u}^{(m)} + (1 - \Theta_{CN}) \mathcal{L}\mathbf{u}^{(m-1)} - \mathcal{G}p^{(m)}] \\ + \alpha_a^{(m)} \mathcal{N}(\mathbf{u}^{(m-1)}) + \alpha_b^{(m)} \mathcal{N}(\mathbf{u}^{(m-2)}) \ , \end{aligned} \tag{5.3.9}$$

where $\mathbf{u}^{(1)} = \mathbf{u}(t + \alpha_a^{(1)} \Delta t)$, $\mathbf{u}^{(2)} = \mathbf{u}(t + (\alpha_a^{(1)} + \alpha_a^{(2)} + \alpha_b^{(2)}) \Delta t)$ are intermediate solutions, and $\mathbf{u}^{(3)} = \mathbf{u}(t + \Delta t)$ is the solution at time $t + \Delta t$. The coefficients $\alpha_a^{(m)}$, $\alpha_b^{(m)}$ and $\alpha_c^{(m)}$, $m = 1, 2, 3$, are listed in Table 5.1.

The parameter Θ_{CN} allows to choose between a fully explicit or a semi-implicit scheme:

- for $\Theta_{CN} = 0$, the (CN-)RK3 scheme (5.3.9) is fully explicit and corresponds to a low-storage, three-stage Runge-Kutta scheme (RK3) of (global) order 3 [4, chapt. 2.3.1].
- for $\Theta_{CN} = 0.5$ (which is the value we use in the code), the (CN-)RK3 scheme (5.3.9) is semi-implicit, where the unconditionally stable Crank-Nicolson scheme (CN) of

Table 5.1: Coefficients of the (CN-)RK3 time integration scheme. Table data and caption taken from [4, table 2.2].

m	$\alpha_a^{(m)}$	$\alpha_b^{(m)}$	$\alpha_c^{(m)}$
1	$\frac{8}{15}$	0	$\frac{8}{15}$
2	$\frac{5}{12}$	$-\frac{17}{60}$	$\frac{2}{15}$
3	$\frac{3}{4}$	$-\frac{5}{12}$	$\frac{1}{3}$

(global) order 2 is used for the integration of the linear term $\mathcal{L}\mathbf{u}$, while the explicit time integration of the nonlinear term $\mathcal{N}(\mathbf{u})$ is performed with the RK3-scheme [4, chapt. 2.3.1]. This scheme allows to "avoid the restrictive viscous time step limit" [6, sect. 4.1].

The scheme (5.3.9) results in the following coupled system of linear equations for the velocity $\mathbf{u}^{(m)}$ and the pressure $p^{(m)}$ of the new sub-time level m

$$\begin{bmatrix} \mathcal{H}^{(m)} & \alpha_c^{(m)}\Delta t\mathcal{G} \\ \mathcal{D} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(m)} \\ p^{(m)} \end{bmatrix} = \begin{bmatrix} q(\mathbf{u}^{(m-1)}, \mathbf{u}^{(m-2)}) \\ 0 \end{bmatrix}, \quad \text{for } m = 1, 2, 3\,, \tag{5.3.10}$$

where

$$\mathcal{H}^{(m)} = 1 - \Theta_{CN}\alpha_c^{(m)}\Delta t\mathcal{L} \tag{5.3.11}$$

is the Helmholtz operator and

$$\begin{aligned} q(\mathbf{u}^{(m-1)}, \mathbf{u}^{(m-2)}) = \big[1 + (1 - \Theta_{CN})\alpha_c^{(m)}\Delta t\mathcal{L}\big]\mathbf{u}^{(m-1)} \\ + \alpha_a^{(m)}\mathcal{N}(\mathbf{u}^{(m-1)}) + \alpha_b^{(m)}\mathcal{N}(\mathbf{u}^{(m-2)}) \end{aligned} \tag{5.3.12}$$

stands for the remainder of Equation (5.3.10).

As mentioned in [4, chapt. 2.3.1], the repeated solution of the linear system (5.3.10) is "typically by far the most time-consuming part of a numerical simulation".

5.4 Spatial discretization scheme

For this section, we follow closely the description and the notation in [4, chapt. 2.3.2].

IMPACT handles Cartesian coordinates and rectangular domains with arbitrary grid stretching [6, sect. 4.2]. Explicit finite differences of high convergence order are used as a local spatial discretization scheme. Based on (5.3.10), this leads to a SLE of the form

$$\begin{bmatrix} \mathbf{H} & \mathbf{G} \\ \mathbf{D} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \mathbf{0} \end{bmatrix}, \tag{5.4.1}$$

which has to be solved in each sub-time step of the time integration scheme (the index m for the sub-time step level is dropped from now on to simplify the notation). The vector $\mathbf{u} = [\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3]^T$ represents the discrete velocity and $\mathbf{p}$ represents the discrete pressure. The matrix $\mathbf{D}$ is the discretized form of the divergence operator $\mathcal{D} = \nabla\cdot(\cdot)$, and the matrix $\mathbf{G}$ is the discretized form of the gradient operator $\alpha_c\Delta t\mathcal{G} = \alpha_c\Delta t\nabla(\cdot)$. The discretized Helmholtz operator $\mathcal{H}$ has the form

$$\mathbf{H} = \mathbf{J} - \frac{1}{2}\alpha_c\Delta t\mathbf{L} \tag{5.4.2}$$

for the semi-implicit time integration scheme, where $\mathbf{L}$ stands for the discretized form of the linear operator $\mathcal{L} = \frac{1}{\mathrm{Re}}\triangle(\cdot)$, and the form

$$\mathbf{H} = \mathbf{J} \tag{5.4.3}$$

for the fully explicit time integration scheme.

The matrix $\mathbf{J}$ is equal to the identity matrix $\mathbf{I}$ except that the rows corresponding to boundary points hold the stencils describing the velocity boundary conditions. The matrices $\mathbf{L}$ and $\mathbf{G}$ act everywhere except on the boundary, i.e. their rows corresponding to boundary points are left blank (these same rows in $\mathbf{J}$ and $\mathbf{q}$ describe the boundary conditions for the velocity). In contrast, the continuity equation is imposed everywhere and $\mathbf{Du} = \mathbf{0}$ acts also on boundary points.

By taking the Schur complement of Equation (5.4.1), an equation for the pressure is obtained, given by

$$\mathbf{DH}^{-1}\mathbf{Gp} = \mathbf{DH}^{-1}\mathbf{q}\,. \tag{5.4.4}$$

Once the pressure is found, the velocity $\mathbf{u}$ can be determined from

$$\mathbf{Hu} = \mathbf{q} - \mathbf{Gp}\,. \tag{5.4.5}$$

5.4.1 Staggered grids

The finite differences stencils are used on staggered grids for the velocity and the pressure. There are four sub-grids (Figure 5.2): one for each velocity component and one for the pressure. The pressure grid is labeled 0 and the velocity grids are labeled 1, 2 or 3 (corresponding to the direction of the velocity component). The momentum equations are solved on the respective velocity grids, and the continuity equation is satisfied on the pressure grid.

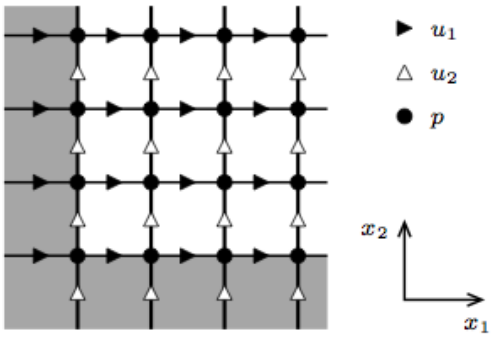


Figure 5.2: Staggered grid in two dimensions near boundaries. Figure and caption taken from [4, fig. 2.2].

The discrete divergence operator $\mathbf{D}$ computes first derivatives on grid 0 from function values stored on grids 1, 2 and 3, whereas the discrete gradient operator $\mathbf{G}$ computes first derivatives on the grids 1, 2 and 3 from function values stored on grid 0. The discrete Laplacian operator $\mathbf{L}$ computes second derivatives directly in the respective velocity grids. It is obtained by applying subsequently $\mathbf{G}$ then $\mathbf{D}$.

For the advective term $(\mathbf{u}\cdot\nabla)\mathbf{u}$, the first derivative on grid i in direction j is represented by the discrete operator

$$\mathbf{C}_{i,j} \approx \frac{\partial(\cdot)_i}{\partial x_j}, \quad i,j = 1,2,3\,. \tag{5.4.6}$$

Additionally, the advection velocities have to be transferred between the velocity grids. The discrete interpolation operators $T_{i,0}$ and $T_{0,j}$ are used for this. They interpolate function values from the pressure grid 0 onto the velocity grid i and function values from the velocity grid j onto the pressure grid 0, respectively. The local velocity component in direction j on grid i is then obtained from

$$\mathbf{u}_{j,i} = \mathbf{T}_{i,0}\mathbf{T}_{0,j}\mathbf{u}_j, \quad i,j = 1,2,3\ . \tag{5.4.7}$$

The final form of the the advective term $(\mathbf{u}\cdot\nabla)\mathbf{u}$ is

$$u_j\frac{\partial u_i}{\partial x_j} \approx \mathrm{diag}\{\mathbf{u}_{j,i}\}\mathbf{C}_{i,j}\mathbf{u}_i = \mathrm{diag}\{\mathbf{T}_{i,0}\mathbf{T}_{0,j}\mathbf{u}_j\}\mathbf{C}_{i,j}\mathbf{u}_i, \quad i,j = 1,2,3\ , \tag{5.4.8}$$

where $\mathrm{diag}\{\mathbf{u}_{j,i}\}$ is a diagonal matrix with the components of $\mathbf{u}_{j,i}$ as diagonal entries.

From now on, we will call *staggered* an operation that computes derivatives or interpolated values on a grid from a function whose values are stored on a different grid (e.g. the operators $\mathbf{D}$, $\mathbf{G}$ and $\mathbf{T}$), and call *collocated* an operation that computes derivatives or interpolated values on a grid from a function whose values are stored on the same grid (e.g. the operators $\mathbf{L}$, $\mathbf{H}$ and $\mathbf{C}$).

5.4.2 Finite Differences stencils

Two methods are implemented to compute the finite difference and interpolation stencil coefficients.

- In the first method, the coefficients are computed directly on the stretched (physical) grid from truncated Taylor series. For sufficiently smooth functions, the *truncation error* with respect to the exact result "typically scales as $O(\Delta x^{n-1})$" for central collocated operations, and "as $O(\Delta x^n)$" for central staggered operations [4, chapt. 2.3.2].

  We remark that the stencil coefficients are obtained by inverting a Vandermonde-like matrix, which is increasingly ill-conditioned with growing n, such that the "accuracy of the stencil coefficients is limited" [6, sect. 6.1.1]. For our simulations, we use a scheme with up to $n = 7$ coefficients.

- In the second method, an invertible, at least twice differentiable mapping $\mathbf{x}(\mathbf{z})$ is used to switch between the physical grid with coordinates $\mathbf{x}$ and an *equidistant* computational grid with coordinates $\mathbf{z}$ on which all spatial operations are performed. While the above stated convergence orders hold on the equidistant grid, the convergence orders are in general reduced on the physical grid [4, chapt. 2.3.2]. However, this approach has the advantage that it "does not introduce any artificial advection or amplification to the discrete operators in case of nonuniform grids" [4, chapt. 2.3.2].

To "provide an anti-aliasing filter for under-resolved flows" [6, sect. 4.2.2], *upwind-biased* finite differences for the discretization of $\mathbf{C}$ are used. There the outermost coefficients on the downwind sides of the stencils are set to zero (Figure 5.3), and the convergence order reduces to $n-2$ [4, chapt. 2.3.2]. This modified scheme "damps the solution especially at high wave numbers" but "does not affect the dispersion properties" [6, sect. 4.2.2]. The damping of high wavenumber modes has a dissipative effect [4, chapt. 2.3.2] and "controls the accumulation of kinetic energy in the large wave numbers" [6, sect. 4.2.2].

In the interior of the domain, the same stencil width n is used for all collocated operators and the same stencil width $n-1$ for all staggered operators, where n is an odd number and

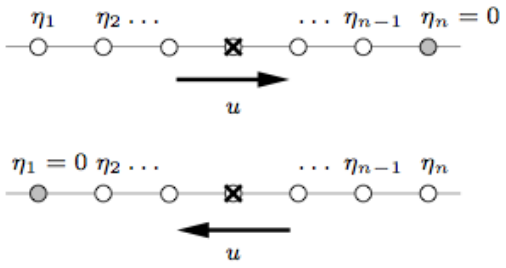


Figure 5.3: Upwind-biased finite-difference stencils, where the η_j are the stencil coefficients. The outermost coefficients on the downwind sides are set to zero. Figure taken from [4, fig. 2.4].

central stencils are used (except for the upwind schemes). Near the boundaries, the stencil widths are reduced and modified stencils are used [4, chapt. 2.3.2].

Five different sets of finite-difference stencils are implemented [4, chapt. 2.3.2], as specified in Figure 5.4. The d3 scheme is the one that we use for our simulations. It is sketched in Figure 5.5.

Name	Truncation error	Grid	Convergence order (number of coefficients)							
d1	$\mathcal{O}(\Delta x^1)$	Colocated	1(2)	1(3)	1(3)	...				
		Staggered	2(2)	2(2)	...					
d2	$\mathcal{O}(\Delta x^2)$	Colocated	2(3)	2(4)	3(5)	3(5)	...			
		Staggered	2(3)	4(4)	4(4)	...				
d3	$\mathcal{O}(\Delta x^3)$	Colocated	3(4)	3(5)	3(5)	5(7)	5(7)	...		
		Staggered	3(4)	4(4)	6(6)	6(6)	...			
d4	$\mathcal{O}(\Delta x^4)$	Colocated	4(5)	4(6)	4(6)	5(7)	7(9)	7(9)	...	
		Staggered	4(5)	4(4)	6(6)	8(8)	8(8)	...		
d5	$\mathcal{O}(\Delta x^5)$	Colocated	5(6)	5(7)	5(7)	5(7)	7(9)	9(11)	9(11)	...
		Staggered	5(6)	5(6)	6(6)	8(8)	10(10)	10(10)	...	

Figure 5.4: Convergence order (and number of non-zero coefficients) of the finite difference stencils on the first few grid points starting from the boundary. The first pair of numbers corresponds to the grid point on the boundary (collocated) or next to the boundary (staggered), cf. Figure 5.5. Table and caption taken from [6, table 3].

Discretization scheme for LES

The differentiation error of the previously described finite difference stencils is "typically most pronounced at high wave numbers" [4, chapt. 2.3.2].

In Large-Eddy Simulations (LES), the differentiation errors become significant ideally only "at wavenumbers which are effectively treated by the SubGrid Scale model" [4, chapt. 2.3.2]. The explicit differentiation schemes described before are however often not sufficiently accurate to achieve this [4, chapt. 2.3.2]. Therefore, compact finite difference schemes (where differentiation schemes are defined implicitly [10, chapt. 3.1.2]) together with the mapping approach (described in Section 5.4.2) are used for LES. The schemes are, for equidistant grids, fourth-order accurate at the boundary and tenth-order accurate in the interior of the domain [4, chapt. 2.3.2]. Since the energy accumulation at high wavenumbers is controlled ideally uniquely by the SubGrid Scale (SGS) model, no "interfering upwind procedure" is employed for the advective terms [4, chapt. 2.3.2].

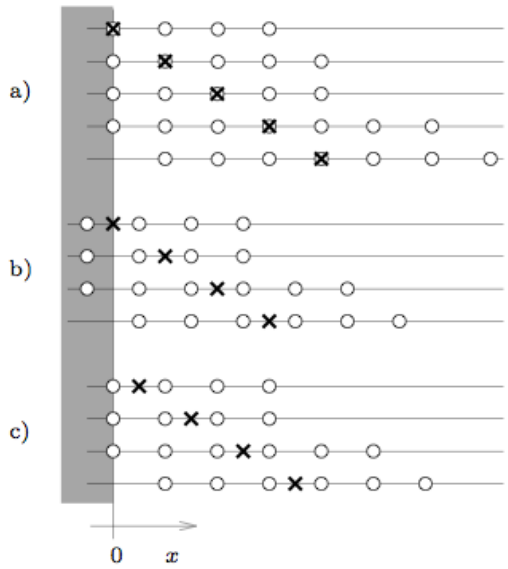


Figure 5.5: Finite difference stencils of the d3 scheme near the boundary. Differentiation scenarios: (a) from a velocity grid to the same velocity grid (collocated operation), (b) from a velocity grid to the pressure grid (staggered operation) and (c) from the pressure grid to a velocity grid (staggered operation). Figure and caption taken from [4, fig. 2.6].

Further discussion of the spatial discretization with compact finite differences schemes is out of the scope of this thesis, and in the following, we assume that explicit finite differences schemes are used. For a short description of one of the SGS models used in the code, we refer to Section 5.8.

5.5 Iterative solution

We recall here that on each sub-time step of the integration scheme, a linear system of the form

$$\begin{bmatrix} \mathbf{H} & \mathbf{G} \\ \mathbf{D} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \mathbf{0} \end{bmatrix} \tag{5.5.1}$$

has to be solved. An equivalent system can be obtained [4, chapt. 2.4.1] by taking the Schur complement of (5.5.1), this leads to

$$\begin{bmatrix} \mathbf{H} & \mathbf{G} \\ 0 & \mathbf{D}\mathbf{H}^{-1}\mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \mathbf{D}\mathbf{H}^{-1}\mathbf{q} \end{bmatrix} . \tag{5.5.2}$$

To solve (5.5.2), an equation for the pressure is solved first

$$\mathbf{A}\mathbf{p} = \mathbf{b} , \tag{5.5.3}$$

where $\mathbf{A} = \mathbf{D}\mathbf{H}^{-1}\mathbf{G}$ and $\mathbf{b} = \mathbf{D}\mathbf{H}^{-1}\mathbf{q}$. Once the pressure is found, the velocity $\mathbf{u}$ can be determined from

$$\mathbf{H}\mathbf{u} = \mathbf{q} - \mathbf{G}\mathbf{p} . \tag{5.5.4}$$

Iterative methods are used to solve the linear systems (5.5.3) and (5.5.4), because "direct solvers have an unfavorable numerical complexity" [6, sect. 5.0] for large problem sizes. Furthermore, iterative methods allow a "direct control of the solution accuracy", which is

useful for problems that do not need high accuracy (e.g. sup-problems appearing in preconditioners) [6, sect. 9.0].

Before discussing the solution of Equations (5.5.3) and (5.5.4), we define the measure

$$\beta \equiv \frac{\Delta t}{2}\|\mathbf{L}\|_\infty \, , \tag{5.5.5}$$

which characterizes the Helmholtz matrix $\mathbf{H}$ [6, sect. 5.1]. For the central discretizations of $\mathbf{L}$ used in the code, it is found [6, sect. 5.1] that

$$\beta \approx \frac{\Delta t}{\mathrm{Re}\, \min\limits_{\Omega}\{\Delta x^2\}} \, . \tag{5.5.6}$$

The measure β can be interpreted as a measure for the number of iterations needed by the solvers to solve the system (5.5.1), independently of the problem size or the degree of parallelization [6, sect. 5.2.0, 8.1.1, 8.1.2]. For very large values of β, we may need to reduce β for the solvers to converge [6, sect. 5.2.1]. Very small values of β indicate (see (5.5.6)) that the time step size is in the viscous time step stability limit and that an explicit time integration may be more efficient (we remark that $\beta = 0$ for the explicit time integration) [6, sect. 5.2.1].

5.5.1 Pressure iteration

With staggered grids it is normally achieved that $\mathbf{A}$ has "normally" a rank-deficit of one (and a corresponding zero eigenvalue) [6, sect. 4.2.1], which accounts for the undefined pressure constant. But since $\mathbf{A}$ has a zero eigenvalue, "typical primary iterative solvers will not work efficiently without an appropriate preconditioner" [6, sect. 5.2.0].

In the code, the preconditioned Richardson iteration scheme is used. It reads, with a preconditioner $\tilde{\mathbf{A}}$:

$$\mathbf{p}^{l+1} = \mathbf{p}^l + \omega \tilde{\mathbf{A}}^{-1}\mathbf{r}_A^l \, , \tag{5.5.7}$$

where l is the iteration count, ω is a relaxation parameter (in the code $\omega = 1$) and $\mathbf{r}_A^l$ is the residual

$$\mathbf{r}_A^l = \mathbf{b} - \mathbf{A}\mathbf{p}^l = \mathbf{D}\mathbf{H}^{-1}(\mathbf{q} - \mathbf{G}\mathbf{p}^l) = \mathbf{D}\mathbf{u}^l \, . \tag{5.5.8}$$

We see from (5.5.8) that the discrete divergence of $\mathbf{u}^l$ is given by the residual $\mathbf{r}_A^l$. The error in the pressure field is

$$\mathbf{e}_A^l = \mathbf{p} - \mathbf{p}^l = \mathbf{A}^{-1}\mathbf{r}_A^l \, . \tag{5.5.9}$$

The Richardson iteration is terminated when the residual satisfies

$$\|\mathbf{r}_A^{l*}\| \leq \varepsilon_A \, , \tag{5.5.10}$$

with the threshold $\varepsilon_A \geq 0$ and the corresponding iteration count l^*. As initial guess $\mathbf{p}^0$, the pressure field from the previous sub-time step is used [6, sect. 5.2.0]. For the very first initial guess, we set the value zero in the code.

Preconditioner

In the code, a commutation-based preconditioner [6, sect. 5.2.0] is used, which has the form

$$\tilde{\mathbf{A}} = \mathbf{D}\mathbf{J}^{-1}\mathbf{G}(\mathbf{D}\mathbf{J}^{-1}\mathbf{H}\mathbf{J}^{-1}\mathbf{G})^{-1}\mathbf{D}\mathbf{J}^{-1}\mathbf{G} \, . \tag{5.5.11}$$

Its application requires two sub-solutions, which are solutions of Poisson problems with matrix $\mathbf{K} = \mathbf{D}\mathbf{J}^{-1}\mathbf{G}$ [6, sect. 5.2.0]. The application of the preconditioner (5.5.11) is illustrated in Figure 5.6.

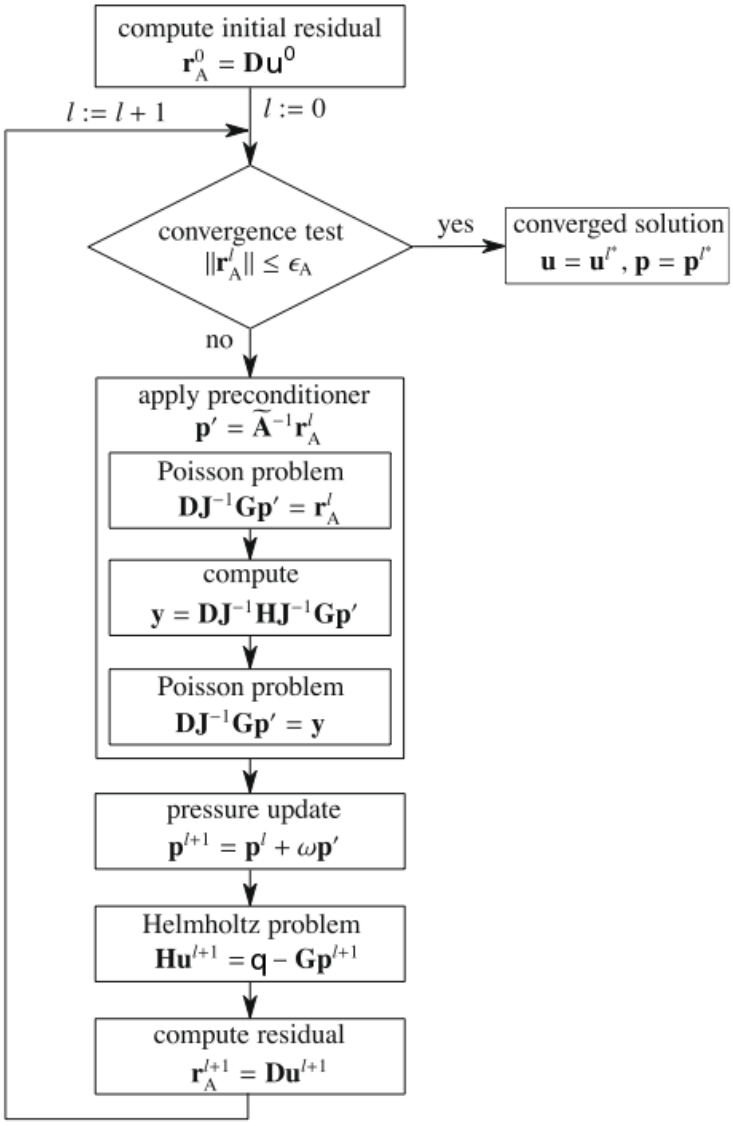


Figure 5.6: Flow chart of the pressure iteration with the preconditioner (5.5.11). The vectors $\mathbf{p}'$ and $\mathbf{y}$ are temporary variables in the context of the preconditioner. Figure and caption taken from [6, fig. 7]. The figure was slightly modified.

The preconditioner $\frac{1}{\omega}\tilde{\mathbf{A}}$ is an "increasingly poor approximation" of $\mathbf{A}$ as β increases, and at a certain point, the Richardson iteration diverges [6, sect. 5.2.1]. This can be avoided, for a given mesh width Δx and Reynolds number Re, by choosing a smaller ω or by reducing the time step size Δt to reduce β [6, sect. 5.2.1].

The complexity to solve the pressure equation is given by the complexity of the Richardson iteration (5.5.7) plus the complexity to apply $\mathbf{D}$ and the preconditioner $\tilde{\mathbf{A}}^{-1}$. And the complexity to apply $\tilde{\mathbf{A}}^{-1}$ is equal to the complexity to apply $\mathbf{K}^{-1} = (\mathbf{D}\mathbf{J}^{-1}\mathbf{G})^{-1}$ plus the complexities of $\mathbf{D}$, $\mathbf{G}$, $\mathbf{H}$ and $\mathbf{J}^{-1}$. The inverse $\mathbf{J}^{-1}$ is "trivial to compute directly since only the boundary conditions need to be inverted" [6, sect. 5.2.0, 5.2.1].

5.5.2 Poisson equations

The preconditioner (5.5.11) includes two Poisson sub-problems of the form

$$\mathbf{K}\mathbf{x} = \mathbf{h} \, . \tag{5.5.12}$$

where the operator $\mathbf{K} = \mathbf{D}\mathbf{J}^{-1}\mathbf{G}$ is of Laplacian-type [6, sect. 5.2.2].

In the code, Equation (5.5.12) is solved with the Krylov subspace method BiCGstab with right preconditioning using geometric multigrid [6, sect. 5.2.2]. That is,

$$\mathbf{x}_{\text{start}} = \tilde{\mathbf{K}}^{-1}\mathbf{h} \tag{5.5.13}$$

is first solved and $\mathbf{x}_{\text{start}}$ is then used as a first guess for the BiCGstab algorithm. The matrix $\tilde{\mathbf{K}}^{-1}$ stands for one application of a geometric multigrid scheme with a $V(m,m)$-cycle (m smoothing sweeps on each grid level) at a grid coarsening factor of two in all spatial directions, where the Gauss-Seidel iteration is used as a smoother [6, sect. 5.2.2]. The fine-grid discretization in the multigrid scheme is the d1 scheme (see Figure 5.4 in Section 5.4.2) [6, sect. 8.1.2]. The restriction is performed by injection ('direct mapping') and the prolongation is performed by bilinear interpolation, which is usually sufficient because "the induced error is normally only a small part of the total approximation error of $\tilde{\mathbf{K}}^{-1}$" [6, sect. 5.2.2].

Equation (5.5.12) is solved iteratively and terminated after $j = j^*$ iterations, when the residual satisfies $\|\mathbf{r}_K^{j*}\| \leq \varepsilon_K$, with the threshold $\varepsilon_K \geq 0$. The threshold at the iteration count $l+1$ of the outer pressure iteration is set to $\varepsilon_K^{l+1} = \phi\|\mathbf{r}_A^{l+1}\|$, with a relaxation factor $\phi < 1$ [6, sect. 5.2.2]. The residual $\mathbf{r}_A^{l+1}$ is extrapolated from the previous time step, such that the termination threshold for the Poisson sub-problems at time t, sub-time step m and iteration $l+1$ [6, sect. 5.2.2] is

$$\varepsilon_{K,t,m}^{l+1} = \phi\|\mathbf{r}_A^{l+1}\|_{t-\Delta t,m} \, . \tag{5.5.14}$$

Values for ϕ between 0.1 and 1.0 (we set the value to 0.5 in the code) are in practice "good choices" [6, sect. 5.5]. They do not need to be smaller because it can "be cheaper overall to tolerate a few more outer pressure iterations (because of a large ϕ) rather than solving the preconditioner problem fewer times but more accurately" [6, sect. 5.5].

For the first Poisson problem, the initial guess is set to zero, and for the second Poisson problem, it is set to the solution of the first Poisson problem [6, sect. 5.2.2]. The number of iterations to solve Equation (5.5.12) with BiCGstab and multigrid is "typically of order one" and does not depend "on the problem size or the degree of parallelization" [6, sect. 5.2.2]. The complexity to compute $\mathbf{K}^{-1}\mathbf{h}$ is then given by the complexities of $\mathbf{K}$, the contributions of the BiCGstab solver and of the multigrid preconditioner [6, sect. 5.2.2].

5.5.3 Helmholtz problem

Once the pressure $\mathbf{p}^l$ is obtained, the system

$$\mathbf{H}\mathbf{u}^l = \mathbf{q} - \mathbf{G}\mathbf{p}^l \tag{5.5.15}$$

can be solved in order to obtain the velocity $\mathbf{u}^l$ and compute the residual $\mathbf{r}_A^l = \mathbf{D}\mathbf{u}^l$ for the next pressure iteration. Once this residual is sufficiently small, a separate solution of the Helmholtz equation (5.5.4) is "usually not necessary" [6, sect. 5.3].

Equation (5.5.15) is solved iteratively and terminated after $k = k^*$ iterations, when the residual

$$\mathbf{r}_H^{l,k} = \mathbf{q} - \mathbf{G}\mathbf{p}^l - \mathbf{H}\mathbf{u}^{l,k} \tag{5.5.16}$$

satisfies $\|\mathbf{r}_H^{l,k}\| \leq \varepsilon_H$, with the threshold $\varepsilon_H \geq 0$. The flow field $\mathbf{u}$ from the previous sub-time step is "usually the best initial guess" [6, sect. 5.3].

The value of β is in practice "sufficiently small" to solve Equation (5.5.15) with the unpreconditioned Krylov subspace method BiCGstab [6, sect. 5.3]. If β is large, the Helmholtz problems tend to be Poisson–like such that they can be treated with multigrid preconditioning, similarly as the Poisson problems (5.5.12) [6, sect. 5.3]. In terms of computational cost, the solution of the Helmholtz problems is "typically equally or less expensive than the application of the preconditioner $\tilde{\mathbf{A}}$" [6, sect. 5.3].

The number of iterations to solve Equation (5.5.15) with BiCGstab to a given level of accuracy depends "mostly on β" but not on the problem size or the degree of parallelization [6, sect. 5.3]. So, the complexity to solve Equation (5.5.15) is given by the complexity of $\mathbf{H}$ and the contributions of the BiCGstab solver [6, sect. 5.3].

Explicit time integration

If the explicit time integration scheme is used, the matrix $\mathbf{H}^{-1}$ reduces to $\mathbf{J}^{-1}$, which is easy to compute directly since "only the boundary conditions need to be inverted" [6, sect. 5.2.1]. Therefore, the Helmholtz problem (5.5.15) can be directly solved, without an iterative solver. Additionaly, the matrix $\mathbf{A}$ and the right-hand side $\mathbf{b}$ reduce to $\mathbf{A} = \mathbf{D}\mathbf{J}^{-1}\mathbf{G}$ and $\mathbf{b} = \mathbf{D}\mathbf{J}^{-1}\mathbf{q}$, such that the pressure solution can be found with the same solver used for the Poisson problems (5.5.12), but with the residual threshold ε_A instead of ε_K. Therefore, the pressure problem (5.5.3) is solved without the outer pressure iteration (5.5.7).

5.5.4 Total error

The error between the numerical solution $\mathbf{u}$ and the exact solution $\mathbf{u}_{\text{exact}}$ can be decomposed in a discretization error $\mathbf{e}_d$ due to the discretization of the operators and an iteration error $\mathbf{e}_{it}$ due to the iterative solution [6, sect. 6.0]. For efficiency reasons, the iteration error "should not be required to be much smaller than the discretization error " [6, sect. 5.4]. Conversely, for the convergence order of the discretization error to be observable, the iteration error should not be bigger than the discretization error.

5.5.5 Solution accuracy

For a given threshold ε_H of the Helmholtz problem, it cannot be expected in general [6, sect. 5.4] that the residual $\|\mathbf{r}_A\|$ of the pressure equation can be reduced below

$$\varepsilon_{A,min} = \sup_{\|\mathbf{r}_H\| \leq \varepsilon_H} \|\mathbf{D}\mathbf{H}^{-1}\mathbf{r}_H\|. \tag{5.5.17}$$

By approximating (5.5.17) (with consistent matrix norms), the following relation is used [6, sect. 5.4] instead

$$\|\mathbf{D}\|\|\mathbf{H}^{-1}\|\varepsilon_H = \varepsilon_A \ . \tag{5.5.18}$$

In the code, we use the relation

$$\varepsilon_A = \varepsilon_H \ , \tag{5.5.19}$$

and the residual norms are calculated in the infinity norm.

5.5.6 Solvability

For the singular system (5.5.1) to have a solution, the right-hand side must be contained in the column space of the system matrix. With such a right-hand side, the rank deficiency of the system matrix is "usually not a problem" for the iterative solvers used in the code [6, sect. 5.6]. As mentioned in [6, sect. 5.6], a right-hand side which is not contained in the column space of the matrix indicates that "the boundary conditions try to enforce a net increase or decrease of mass in the domain [respectively an artificial inflow (or outflow) over the boundaries] which violates the mass conservation law".

In [6, sect. 5.6], two methods to resolve this problem are described. The first one modifies the system matrix by prescribing the pressure artificially at at least one point in space, such that $\mathbf{A}$ becomes "non-singular and a solution always exists" [6, sect. 5.6]. The disadvantage is that the governing equation $\mathbf{Du} = \mathbf{0}$ is replaced at these grid points by an artificial pressure constraint, such that the flow field is "generally not divergence-free there" [6, sect. 5.6]. These points can be interpreted as "mass sinks (or sources) which compensate for the net outflow (or inflow) over the boundaries" [6, sect. 5.6]. The solution is also "normally not smooth" in these areas which can lead to "stability problems during the time integration" [4, chapt. 2.4.6].

In the second method, used in the code [6, sect. 5.6], it is the right-hand side $\mathbf{q} = \mathbf{HD}^{-1}\mathbf{b}$ that is corrected to $\mathbf{q}_{corr} = \mathbf{HD}^{-1}\mathbf{b}_{corr}$, such that the corresponding corrected right-hand side $\mathbf{b}_{corr}$ lies in the column space of $\mathbf{A}$ and the system $\mathbf{Ap} = \mathbf{b}_{corr}$ admits a solution. Once a solution for the pressure is found, the arbitrarily pressure constant can be chosen arbitrarily [6, sect. 5.6].

We describe next how the right-hand side $\mathbf{q}_{corr}$ is corrected, following the explanation in [4, chapt. 2.4.6]. Let the vector $\boldsymbol{\Psi} \neq \mathbf{0}$, with $\boldsymbol{\Psi}^T\mathbf{A} = 0$, represent the left nullspace of $\mathbf{A}$. The vector $\boldsymbol{\Psi}$ can be calculated with the same methods as used for solving the pressure equation (5.5.3) [4, chapt. 2.4.6]. The corrected right-hand side $\mathbf{b}_{corr}$ must then be orthogonal to $\boldsymbol{\Psi}$ since

$$\boldsymbol{\Psi}^T\mathbf{b}_{corr} = \boldsymbol{\Psi}^T\mathbf{Ap} = \mathbf{0} \,. \tag{5.5.20}$$

Let us now define the vector

$$\boldsymbol{\phi} \equiv \mathbf{H}^{-T}\mathbf{D}^T\boldsymbol{\Psi} \,. \tag{5.5.21}$$

The vector $\boldsymbol{\phi}$ can be calculated from $\boldsymbol{\Psi}$ with the same methods as used for solving the Helmholtz problem (5.5.4) [4, chapt. 2.4.6]. The right-hand side $\mathbf{q}_{corr}$ of the Helmholtz problem (5.5.4) must be orthogonal to $\boldsymbol{\phi}$, because

$$\boldsymbol{\phi}^T\mathbf{q}_{corr} = \boldsymbol{\Psi}^T\mathbf{DH}^{-1}\mathbf{q}_{corr} = \boldsymbol{\Psi}^T\mathbf{b}_{corr} = \mathbf{0} \,. \tag{5.5.22}$$

To satisfy the condition (5.5.22), $\mathbf{q}$ is corrected to $\mathbf{q}_{corr}$ by projecting it along a vector onto the orthogonal space to $\boldsymbol{\phi}$, i.e.

$$\mathbf{q}_{corr} = \mathbf{q} - \frac{\boldsymbol{\phi}^T\mathbf{q}}{\boldsymbol{\phi}^T\boldsymbol{\theta}}\boldsymbol{\theta}, \quad \text{for some } \boldsymbol{\theta} \text{ with } \boldsymbol{\phi}^T\boldsymbol{\theta} \neq \mathbf{0} \,. \tag{5.5.23}$$

Then $\boldsymbol{\phi}^T\mathbf{q}_{corr} = \mathbf{0}$ and $\boldsymbol{\Psi}^T\mathbf{b}_{corr} = \mathbf{0}$ are satisfied and Equations (5.5.1) and (5.5.3) have at least one solution [4, chapt. 2.4.6]. The projection vector $\boldsymbol{\theta}$, also called *flux correction vector*, can be chosen freely as long as it satisfies $\boldsymbol{\phi}^T\boldsymbol{\theta} \neq \mathbf{0}$. Loosely speaking, the flux correction vector corrects the accumulation of discretization errors by enforcing the fluxes at the boundaries to sum up to 0 (mass conservation). For example, in a channel flow, the flux correction vector should be zero at the walls and have, e.g. non-zero components at the outflow boundary in the outflow direction. Otherwise, it is possible to choose that the 2-norm of the correction, $\|\mathbf{q}_{corr} - \mathbf{q}\|_2$, is minimal, and $\boldsymbol{\theta} = \boldsymbol{\phi}$ is set for this [4, chapt. 2.4.6]. Fluxes are then corrected on the entire boundary.

Since the vectors $\boldsymbol{\phi}$ and $\boldsymbol{\Psi}$ depend on the matrix $\mathbf{H}$, which is different for the three sub-time steps $m = 1, 2, 3$ and changes with the time step size Δt, for the semi-implicit time integration, these vectors have to be stored separately at each sub-time step and recomputed as soon as the time step size changes [4, chapt. 2.4.6]. In the code, the time step size is changed every n_{TS} time steps (we use $n_{TS} = 10$). For explicit time integration, the vectors $\boldsymbol{\phi}$ and $\boldsymbol{\Psi}$ are unique for all times and all sub-time steps [4, chapt. 2.4.6].

5.6 Computational and communication complexity

We recall that to advance the solution in time by a time step size Δt, the system

$$\begin{bmatrix} \mathbf{H} & \mathbf{G} \\ \mathbf{0} & \mathbf{D}\mathbf{H}^{-1}\mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \mathbf{D}\mathbf{H}^{-1}\mathbf{q} \end{bmatrix} \tag{5.6.1}$$

has to be solved three times.

Following the description in [6, sect. 3.2], it is assumed in this section that the computational domain has N^d grid points, and that a torus network of dimension d is used. The domain is distributed to P processors, such that each sub-domain contains $\frac{N^d}{P}$ grid points.

As mentioned in Section 5.5, the number of iterations to solve Equation (5.6.1) by the iterative algorithms is mainly dependent on β, but not on the problem size or on the degree of parallelization. That is, the computational work of these iterative algorithms is mainly governed by $O(1)$ sparse matrix-vector multiplications. The computational complexity to solve Equation (5.6.1) is then governed by the application of d differentiation stencils of length n to the $\frac{N^d}{P}$ data points in the sub-domain, such that the computational cost to advance the solution by one time step size Δt is $O(dn\frac{N^d}{P})$ [6, sect. 3.2].

Only the ghost cells need to be communicated to the $P^{\frac{1}{d}}$ neighboring sub-domains, such that the communication complexity is given by the product of the d stencils of width n with the surface area of each sub-domain, times the number of neighboring sub-domains. In the best case (neighboring sub-domains are mapped to neighboring processors), the communication cost is $O(dn\frac{N^{d-1}}{P}P^{\frac{1}{d}}) = O(dn(\frac{N^d}{P})^{\frac{d-1}{d}})$. In the worst case (neighboring sub-domains are mapped to pairs of processors separated by a distance $P^{\frac{1}{d}}$, the communication cost is $O(dn(\frac{N^d}{P})^{\frac{d-1}{d}}P^{\frac{1}{d}})$ [6, sect. 3.2].

Multigrid on parallel processors adds a cost of $O(d\ \log P) + O(P^{\frac{1}{d}})$ [6, sect. 3.2], but it plays a secondary role compared to the communication cost of the ghost cell updates on the fine-grid mesh [6, sect. 8.1.2].

5.7 (Non-exhaustive) list of parameters that can be set in IMPACT

- Initial velocity field (on velocity grid) and initial guess for the very first pressure iteration (on pressure grid), usually zero.
- Type of the boundary conditions for the velocity: symmetry, periodic, Dirichlet or advective.
- Use of the flux corrections. If used, choice of flux correction vector on the boundaries of choice of minimal 2-norm.
- Extents L_1, L_2, L_3 of the domain.

- Number of grid points in each direction (on the grid 0) of the form $M_i = a2^l + 1$. $M_3 = 2$ is set for 2D simulations. With this value of M_3, the code switches off all computations in the third direction.
- Number of sub-domains $\texttt{NB}_i$ in each direction with $\texttt{mod}(M_i - 1, \texttt{NB}_i) = 0$. $\texttt{NB}_3 = 1$ is set for 2D simulations.
- Mapping $\mathbf{x}(\mathbf{z})$ where $\mathbf{z}$ is the computational grid.
- Reynolds number Re.
- Start and end time for the time integration.
- Maximal number of time steps.
- Time integration scheme: semi-implicit or explicit.
- Maximal time step size.
- Maximal time step size for the first `Int_dtime` time steps of the time integration.
- Number of time steps n_{TS} after which the time step size is recomputed.
- Normalized CFL-number $\tilde{\text{CFL}}$.
- Use of upwind scheme.
- Use of mapping for computing the finite differences.
- Use of compact finite differences, and in that case, for which discrete operators.
- Use of LES and LES parameters.
- Maximal number of iterations for the Richardson, the Poisson and the Helmholtz problems.
- Residual threshold ε_H for the Helmholtz iteration (we use $\varepsilon_A = \varepsilon_H$ for the Richardson iteration).
- Ratio $\frac{\varepsilon_K}{\varepsilon_A}$. It is set to 0.5 for our simulations.
- Number of relaxation sweeps in the multigrid. It is set to 4 for our simulations.
- Choice of Gauss-Seidel or Jacobi smoothing in the multigrid.
- Settings for outputs.
- Volume forces in the momentum equation.
- Dirichlet boundary conditions. No need to specify them if they are the same as in the initial condition.
- Values of $c(\mathbf{x})$ (on the pressure grid) and $\mathcal{A}(\mathbf{x}, t)$ for the advective boundary conditions.

5.8 Turbulence modeling in IMPACT

The turbulence models implemented in IMPACT are of LES type and are the high-pass filtered Smagorinsky model and the so-called *ADM-RT* (Approximate Deconvolution Model - Relaxation Term). We summarize here very briefly the *ADM-RT* model, from the description given in [4, chapt. 5.2]. Further discussion of this model is out of the scope of this thesis.

In LES, a *filtered* velocity field $\bar{\mathbf{u}} = \mathcal{F}\mathbf{u}$, where $\mathcal{F}$ is a spatial low-pass filter, is solved [4, chapt. 5.1]. The filter $\mathcal{F}$ used is the so-called *implicit* grid filter. This filter commutes with differentiation in the continuous equations [4, chapt. 5.1], so by applying it to the Navier-Stokes equation one obtains

$$\nabla \cdot \bar{\mathbf{u}} = 0 \tag{5.8.1}$$

$$\partial_t \bar{\mathbf{u}} + (\bar{\mathbf{u}} \cdot \nabla)\bar{\mathbf{u}} = -\nabla \bar{p} + \mathrm{Re}^{-1} \triangle \bar{\mathbf{u}} + \bar{\mathbf{f}} + \mathbf{s} \,, \tag{5.8.2}$$

where $\mathbf{s}$ is the *SGS* (SubGrid Scale) term and can be written as $s_i = -\frac{\partial}{\partial x_j}(\overline{u_i u_j} - \bar{u}_i \bar{u}_j) = -\frac{\partial}{\partial x_j}\tau_{ij}^R$, where τ_{ij}^R is the *residual stress tensor* [14, chapt. 13.3.1]. This SGS-term cannot be obtained by the equations themselves and has to be modeled (closure problem) [14, chapt. 13.3.1].

Here, $\mathbf{s}$ is modeled by $\mathbf{s} \approx -\chi \mathbf{F}_{hp}\mathbf{u}$, $\chi \geq 0$ [4, chapt. 5.2.1], and in the code this term is simply added as an additional term.

$\mathbf{F}_{hp}$ is a high-pass filter of the form $\mathbf{F}_{hp} = (\mathbf{I} - \mathbf{F}_{lp}^{M_{lp}})^{M_{hp}}$ where $\mathbf{F}_{lp}$ is a low-pass filter that we obtain by applying subsequently one-dimensional filters: $\mathbf{F}_{lp} = \mathbf{F}_{lp_1}\mathbf{F}_{lp_2}\mathbf{F}_{lp_3}$ [4, chapt. 5.2.1].

The stencils of width n for the filters $\mathbf{F}_{lp_k}$, $k = 1, 2, 3$ at a point $\mathbf{x}_0$ are

$$\eta_i = \{\mathbf{B}^{-1}\}_{i,1},\ i = 1, \ldots, n \,, \tag{5.8.3}$$

where $\mathbf{B}$ is given by

$$\{B_{i,j}\} = \{(x_0 - x_j)^{i-1}\},\ i = 1, \ldots, n-1,\ j = 1, \ldots, n \,, \tag{5.8.4}$$

$$\{B_{n,j}\} = \{(-1)^{-j}\},\ j = 1, \ldots, n \,, \tag{5.8.5}$$

where x_0 is the kth coordinate $\mathbf{x}_{0,k}$ of the point $\mathbf{x}_0$ [4, chapt. 5.2.1].

On equidistant grids, $\mathbf{F}_{hp}$ and $\mathbf{F}_{lp}$ are symmetric positive (semi-)definite, such that $\bar{\mathbf{u}} \cdot \mathbf{s} \leq 0$, which ensures that the model dissipates energy [4, chapt. 5.2.2]. Stretched grids and asymmetric filter stencils at the boundaries usually "do not affect this property" [4, chapt. 5.2.2].

This concludes the chapter about the IMPACT code. In our implementation of the MLMC algorithm, we will use this solver to calculate the pathwise solutions $S_\ell(t, 0)\mathbf{u}_0$ needed to compute the values Φ_ℓ. Next, we describe how we implemented the MLMC algorithm together with the IMPACT Finite Differences solver, in a parallel environment and using a static load balancing strategy, and call it simply 'MLMC-FD'.

Chapter 6

MLMC-FD solver

We describe here the implementation in parallel of a MLMC-FD (multilevel Monte Carlo - Finite Differences) solver, based on a static load balancing strategy. We follow for this the lines of [17, sect. 2.4], where a highly scalable implementation for Finite Volume solvers has been tested and validated.

6.0.1 Static load balancing

In the multilevel Monte Carlo method, we approximate the generalized moment $\mathbb{E}_{\mu_t}(\Phi)$ by calculating the statistical estimator

$$E^L_{\mu_t}(\Phi_L) = \sum_{\ell=0}^{L} E^{M_\ell}_{\mu_t}(\Phi_\ell - \Phi_{\ell-1}) \ , \tag{6.0.1}$$

where we used $\Phi_{-1} \equiv 0$ and where $E^{M_\ell}_{\mu_t}(\cdot)$ stands for the Monte Carlo sample mean using M_ℓ samples.

As mentioned in [17, sect. 2.3], there are three possible degrees of parallelization for the calculation of the estimator in (6.0.1): across the discretization levels ℓ, across the Monte Carlo samples M_ℓ and inside the deterministic FD solver that computes Φ_ℓ, using domain decomposition. We assume a *homogeneous computing environment* (i.e. "all cores have identical CPUs and RAM per node, and equal bandwidth and latency to all other cores" [17, sect. 2.3]), and assign for each level $0 \leq \ell \leq L$ a number $C_\ell = P_\ell D_\ell$ of cores, where D_ℓ stands for the number of sub-domains used by the FD solver and P_ℓ stands for the number of *samplers*, which are groups of cores that compute some portion of the M_ℓ Monte Carlo samples at level ℓ [17, sect. 2.3].

Estimation of the computational work

As seen in Chapter 4, Section 4.2, in order to equilibrate statistical and spatio-temporal discretization errors, we use the following relation for the sample numbers M_ℓ (the term $M_0 = O\big((\frac{\nu}{h_L^\sigma})^2\big)$ is here omitted for simplicity)

$$M_\ell = M_L(\frac{h_\ell}{h_L})^{2\sigma} = M_L 2^{2(L-\ell)\sigma} \ , \tag{6.0.2}$$

with the convergence rate $\sigma > 0$ and the (small) number of samples M_L on the finest resolution level. We have assumed here for simplicity that the meshwidth h_ℓ on the discretization level ℓ is equal to $2^{-\ell}$.

The computational complexity of the FD solver IMPACT to compute the solution of a two-dimensional problem on a grid with meshwidth h_ℓ and after a number $O(\Delta_\ell t^{-1})$ of time steps is $O(h_\ell^{-2}) \cdot O(\Delta_\ell t^{-1})$, because iterative solvers with sparse matrix-vector multiplications are used to solve Helmholtz and/or Poisson problems (see Chapter 5, Section 5.6). Also, by using a semi-implicit time integration scheme, we may write $\Delta_\ell t = O(h_\ell)$, such that the computational work to compute M_ℓ Monte Carlo samples can be estimated by

$$\text{Work}_{M_\ell}(h_\ell) = M_\ell \cdot O(h_\ell^{-3}) \ , \tag{6.0.3}$$

and the computational work to calculate the value $E_{\mu_t}^{M_\ell}(\Phi_\ell - \Phi_{\ell-1})$ at level ℓ can then be estimated by

$$\text{Work}_\ell \equiv \text{Work}_{M_\ell}(h_\ell) + \text{Work}_{M_\ell}(h_{\ell-1}) \ . \tag{6.0.4}$$

For an efficient computation of the MLMC estimator $E_{\mu_t}^{L}(\Phi_L)$ in parallel, the cost to compute the sample means $E_{\mu_t}^{M_\ell}(\Phi_\ell - \Phi_{\ell-1})$ should be about the same on all parallel levels ℓ. Assuming weak scalability of the solver, this leads to the relation

$$\frac{\text{Work}_{\ell+1}}{C_{\ell+1}} \equiv \frac{\text{Work}_\ell}{C_\ell} \ . \tag{6.0.5}$$

Assuming a bound (6.0.3) where lower order terms in h_ℓ can be neglected and taking into account (6.0.2), the following relation for the number of cores C_ℓ on level ℓ was derived in [17, sect. 2.3]

$$C_\ell = \text{ceil}\Big(\frac{C_{\ell+1}}{2^{3-2\sigma}}\Big), \quad \forall \ell < L \ , \tag{6.0.6}$$

with the number of cores $C_L = P_L D_L$ on the finest level L being fixed. In the case $\sigma < 1.5$, we see that we have an inefficient load balancing for levels $\ell \leq \ell^*$, where $C_{\ell^*} < 1$. Assuming that P_L and D_L are powers of 2, it is shown in [17, sect. 2.3] that an efficient load balancing can be obtained in this case by assigning multiple levels $\ell = 0, \ldots, \ell^*$ to *one single core*. This is even *essential* in order to obtain an "efficient and highly scalable parallelization" [17, sect. 2.3]. An example for a static load balancing distribution for MLMC-FD can be seen in Figure 6.1.

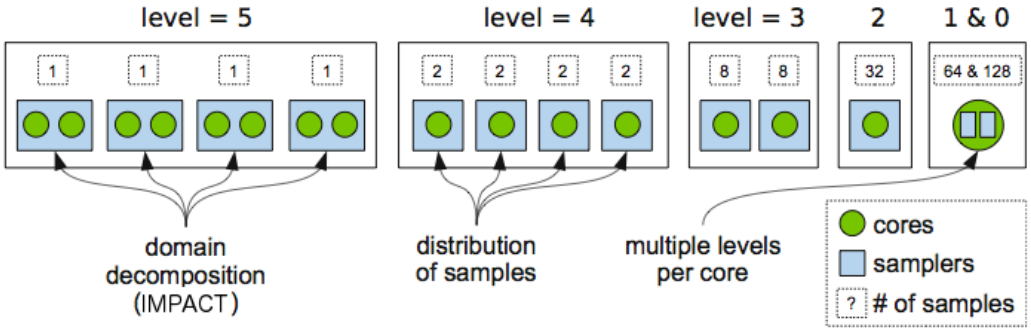


Figure 6.1: Static load balancing structure: $L = 5, M_L = 4, D_L = 2, P_L = 4$. Figure and caption taken from [17, fig. 1]. The figure was slightly modified.

We remark that the above estimation (6.0.6) is to be considered carefully in the IMPACT code. One of the reasons is that, since we assumed that $\Delta_\ell t = O(h_\ell)$, the parameter $\beta \approx \nu \frac{\Delta_\ell t}{h_\ell^2} = \nu O(h_\ell^{-1})$, which can be interpreted as a measure for the convergence speed of the iterative solvers (see Chapter 5, Section 5.5), increases with decreasing h_ℓ, such that in a finer discretization level, for a given level of accuracy, we have not only to integrate the

solution over more time steps, but also each time step takes longer. We do not believe this changes the ratio $\frac{\text{Work}_{M_{\ell_1}}(h_{\ell_1})}{\text{Work}_{M_{\ell_2}}(h_{\ell_2})}$ significantly for ℓ_1 close to ℓ_2, but it may in the other case. More importantly, we assumed before that we had an *homogeneous computing environment* with equal bandwidth and latency between all cores, but this is unfortunately not always the case. This can have a non-negligible influence when the cores assigned for the domain decomposition in the FD solver are not all on the same computing node (in which the cores benefit from very fast inter-connections). Indeed, while the computations in the IMPACT code scale weakly for a given β (as measured in [6, sect. 8.1.2]), the communication costs due to the ghost-cell updates generally do not in case of uneven networking [6, sect. 8.1.2].

Therefore, we found that for the IMPACT solver, the best processor assignment may be problem specific and benchmarking may be advisable, and because of communication costs, the relation (6.0.6) for the processor assignments is more to be interpreted as a 'rule of thumb'. Still, it is important that the implementation can handle the case of multiple levels $\ell = 0, \dots, \ell^*$ on one core.

6.1 Implementation

The IMPACT code is written in FORTRAN90 and uses the *Message Passing Library* (MPI) for communication. The *Hierarchical Data Format* (HDF5) is used for parallel I/O of large datasets [4, chapt. 2.5]. We implemented the MLMC-FD algorithm in FORTRAN90 on top of the IMPACT code (i.e. we extended or modified some of the source files), using also MPI, by following the implementation guidelines from [17, sect. 2.4]. We will summarize the method next. But first, we describe shortly the main steps during a simulation with the IMPACT code and their importance for our implementation, as well as the random number generator (RNG) that we use to generate samples for the random initial velocities.

6.1.1 Workflow of the IMPACT code

The execution of IMPACT is mainly divided into two steps: initialization and time integration. We assume that we have assigned the correct number of cores such that the program can perform a given domain decomposition.

Initialization

During the initialization, input parameters (such as domain size, Reynolds number, ...) are read and tested, then the domain decomposition is performed, i.e. the cores are assigned to sub-domains (according to the partitioning inputted by the user) and a cartesian MPI communicator `COMM_CART` is created from the main communicator `MPI_COMM_WORLD`. The `COMM_CART` communicator as well as communicators derived from it are used for all communications between sub-domains, for example for the synchronization of ghost cells. Later on, to create the communicator `COMM_CART`, we will not use the default communicator `MPI_COMM_WORLD`, which connects all processes, but we will use a communicator `comm_domain`, which connects only a fraction of all available processes. This allows to run the IMPACT code in parallel during the same simulation.

Once the domain decomposition is set up, helping variables and running indices are created, and arrays holding portions of the discrete operators and global vectors are allocated dynamically into the processes memory. Finally, the grid coordinates are calculated and the coefficients for the Finite Differences and interpolation stencils are calculated and tested. This ends the initialization step.

Time integration

The time integration is then performed by calling a routine `timeintegration`. Before calling this routine, parameters that affect the spatial discretization, e.g. the problem size, the domain decomposition, the type of boundary conditions or the grid geometry, cannot be changed. The fact that the problem size cannot be changed indicates that for the calculation of

$$\frac{1}{M_\ell}\sum_{i=1}^{M_\ell}\left(\Phi(S_\ell(t,0)\mathbf{v}^i)-\Phi(S_{\ell-1}(t,0)\mathbf{v}^i)\right), \tag{6.1.1}$$

a group of processors should not calculate consecutively the discrete solutions $S_\ell(t,0)\mathbf{v}^i$ and $S_{\ell-1}(t,0)\mathbf{v}^i$ for a given i on the different discretization levels ℓ and $\ell-1$, because this would imply performing the whole initialization again.

Other parameters, such as the initial condition or settings for the iterative solvers, can be changed just before the time integration. Later on, we will use this property to put the call to the routine `timeintegration` into a loop and assign a different (random) initial condition in each iteration. The routine `timeintegration` implements a measurement of the running time, based on the FORTRAN90 routine `DATE_AND_TIME`. We will later use these measurements to estimate the cost due to the time integration.

6.1.2 Pseudo random number generation

For the generation of (pseudo) uniformly distributed random numbers, a robust random number generator (RNG) is needed, because, as mentioned in [17], "inconsistent seeding and insufficient period length of the RNG might cause correlations in presumably i.i.d. draws which might potentially lead to biased solutions". We used an implementation in FORTRAN90 of the Mersenne-Twister `MT19937` RNG from [8]. The generated numbers are in $(0,1)$ and have 52 bits accuracy [8].

6.1.3 MLMC-FD

We aim here at implementing a scalable parallel application for the computations of MLMC estimators for the mean velocity $\mathbb{E}_{\mu_t}(\mathbf{u})$ or for generalized moments $\mathbb{E}_{\mu_t}(\Phi(\mathbf{u}))$. We use a static load balancing strategy, where we assign $C_\ell = P_\ell \times D_\ell$ cores for the computation of the Monte Carlo sample means $E^{M_\ell}_{\mu_t}(\cdot)$, for $\ell = 0,\dots,L$. These cores are divided into P_ℓ groups of D_ℓ cores (as already mentioned, we call this groups also 'samplers'), and each of these groups computes with the IMPACT solver $\frac{M_\ell}{P_\ell}$ Monte Carlo samples, where the D_ℓ cores are used for the domain decomposition in IMPACT.

We divide the simulation into three phases: initialization, simulation and data collection. We assume next that each MPI process runs on its own core.

Initialization

Creation of communicators: In MPI, different parallel processes can communicate with each other when they belong to the same group, or *communicator*. Inside such a communicator, each process becomes a unique identifier called *rank*, which is a non-negative integer (we remark that this identifier is unique only *within* the communicator). The process with the rank 0 is called *root* of the communicator. The main communicator in MPI is `MPI_COMM_WORLD`, which is created by default and connects all processes of the running application.

In our implementation, an MPI process belongs to three or four of the following types of communicators (besides `MPI_COMM_WORLD`):

i `comm_entire_level`, which connects all C_ℓ processes inside the level ℓ. It is created from `MPI_COMM_WORLD` based on the rank in `MPI_COMM_WORLD` and the partitioning $(C_\ell, \ell = 0, \ldots, L)$ of processes per level.

ii `comm_domain`, which connects D_ℓ processes (these D_ℓ processes form a 'sampler'), that will be used for the domain decomposition in the IMPACT solver. It is created from `comm_entire_level` based on the rank in `comm_entire_level` and on the number D_ℓ. A process with rank 0 in `comm_domain` is called *domain root*.

iii `comm_samplers`, which connects corresponding sub-domains between the P_ℓ samplers. It is created from `comm_entire_level` based on the rank of the sub-domain in `comm_entire_level`. A process with rank 0 in `comm_samplers` is called *sampler root*.

iv `comm_level_roots`, which connects processes that are roots in both `comm_domain` and `comm_samplers`. It is created from `MPI_COMM_WORLD`. The process with rank 0 belongs to the finest level $\ell = L$.

For the creation of subgroups and communicators, we use the `MPI` functions `MPI_Group[_range]_incl()` and `MPI_comm_create()`. In Figure 6.2 we can see the structure of the communicators for the setup as in Figure 6.1.

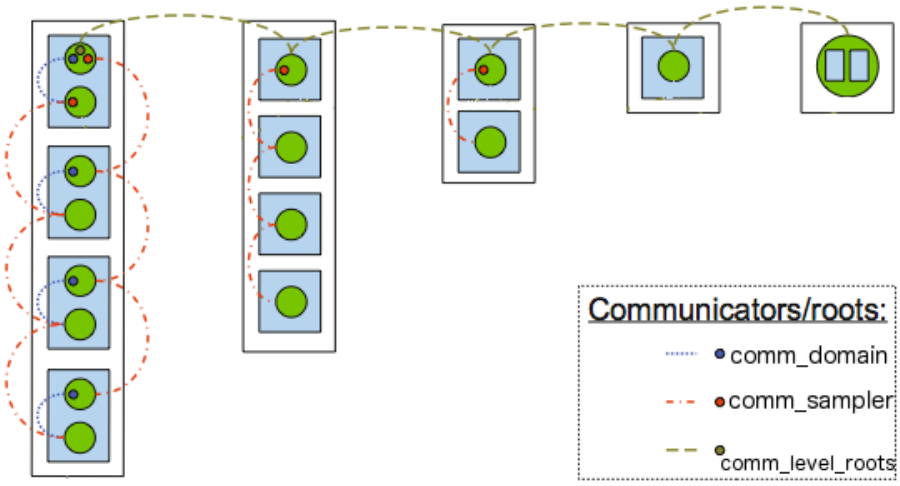


Figure 6.2: Structure and root processes of the communicators for the setup depicted in Figure 6.1. Figure and caption taken from [17, fig. 1]. The figure was slightly modified.

Random number generation: On each level ℓ, we need to generate $M_\ell \times \kappa$ random samples, where κ is the number of samples needed to generate the initial condition. To generate this sequence of random numbers, we use for the RNG a seed based on the level number ℓ (and the simulation number if we do many independent simulations of the same problem).

In our implementation, each process assigned to level ℓ generates the full sequence of $M_\ell \times \kappa$ random samples, and uses some portion of length $\frac{M_\ell}{P_\ell} \times \kappa$ for the simulation. For our purposes, the number of samples $M_\ell \times \kappa$ was not so big (up to about 20000) such that it could become an issue. Else, it is also possible with our RNG to generate P_ℓ independent streams of only $\frac{M_\ell}{P_\ell} \times \kappa$ samples for a given seed [8], and then each core assigned to level ℓ and to sampler $j \in \{1, \ldots, P_\ell\}$ generates such a stream instead of the longer sequence.

In all cases, generating the full sequence of samples only on domain and sample roots and then scattering or broadcasting samples via the domain or sampler communicators "should be avoided", because it introduces "unnecessary communication and memory overheads" [17, sect. 2.4].

Simulation

FD solves: Each group of D_ℓ processes (or 'sampler') inside a domain communicator does $\frac{M_\ell}{P_\ell}$ FD solves (one for each initial random sample $\mathbf{v}^i$) with the IMPACT solver, calculating therefore a part of $E^{M_\ell}_{\mu_t}(\Phi_\ell)$ and a part of $E^{M_\ell}_{\mu_t}(\Phi_{\ell-1})$. In our implementation, we do first all $\frac{M_\ell}{P_\ell}$ FD solves for $E^{M_\ell}_{\mu_t}(\Phi_{\ell-1})$ and then all $\frac{M_\ell}{P_\ell}$ FD solves for $E^{M_\ell}_{\mu_t}(\Phi_\ell)$, to avoid multiple reinitializations of the IMPACT code.

Data Collection

MC estimator: In each level ℓ, the sub-domains between different samplers collectively reduce their part of $E^{M_\ell}_{\mu_t}(\Phi_\ell)$ and of $E^{M_\ell}_{\mu_t}(\Phi_{\ell-1})$ to $E^{M_\ell}_{\mu_t}(\Phi_\ell)$ and $E^{M_\ell}_{\mu_t}(\Phi_{\ell-1})$ into sub-domains of the sampler root, with `MPI_Reduce()`. In our implementation, we compute also the MC means $E^{M_\ell}_{\mu_t}(\mathbf{u}_\ell)$ and $E^{M_\ell}_{\mu_t}(\mathbf{u}_{\ell-1})$ for the velocity fields. Both of these mean vector fields are outputted by the sampler roots (using the HDF5 library) already at this step.

MLMC estimator: Then, $E^{M_\ell}_{\mu_t}(\Phi_\ell)$ and $E^{M_\ell}_{\mu_t}(\Phi_{\ell-1})$ are combined into the MC estimators $E^{M_\ell}_{\mu_t}(\Phi_\ell - \Phi_{\ell-1})$ on domain and sampler roots, and these estimators are then combined via the `comm_level_roots` communicator into the domain and sampler root on level $\ell = L$. Finally, this domain and sampler root on level $\ell = L$ outputs the result.

This concludes the description of the MLMC-FD solver we implemented. Next we describe the machine on which the solver was run for our numerical experiments, and present the results of these experiments in the next chapter.

6.2 Computing resources

The code was run on a machine called 'Pilatus' at the Swiss National Computing Center in Lugano [13].

6.2.1 Description of the machine

We summarize here the informations given in [13].

Pilatus is an Intel SandyBridge cluster composed of 44 computing nodes. Each node has 2 × 8-core Intel(R) Xeon(R) CPU E5-2670 @ 2.60GHz, and 64GB DDR3 memory. The 16 physical cores have Hyper-Threading enabled, such that a "pure MPI job" can actually ask up to 32 MPI tasks per node [13]. Two nodes provide login facilities for user access and compilation, such that the maximum number of virtual cores that can be required is 1344. The maximum allowed running time is 24h and the maximum number of running jobs per user is 3.

6.2.2 Programming environment

The operating system is SUSE SLES11.2 [13]. The programming environment we used was pgi/12.5 from the Portland Group. It loads the mvapich2/1.8 MPI library. As written in

[15], "MVAPICH and MVAPICH2 are high-performance implementations of the Message Passing Interface (MPI) standard which run over InfiniBand interconnects".

Compilation

For the compilation of the IMPACT code, we used the pgf90 compiler from pgi/12.5. The IMPACT uses HDF5 for parallel I/O, such that we also loaded the hdf5/1.8.9 library.

Floating point operations in double precision (8 bytes) were ensured with the compiler flag `-r8`, while integers are treated as 4 bytes variables with the flag `-i4`.

Chapter 7

Results

In this chapter, we describe the results from numerical experiments that were done to test the validity and the feasibility of the MLMC-FD method. There are not related to a specific physical scenario.

7.1 Common setup

We restrict ourselves to the space dimension $d = 2$, and consider the Navier-Stokes equations (1.3.2) with stochastic initial data, described by a given probability distribution μ_0. We assume that there are no volume forces, i.e. $\mathbf{f} = \mathbf{0}$.

The domain is set to $D = (0,1) \times (0,1)$, and we consider the case of periodic boundary conditions with vanishing space average. The spaces H and V correspond then to $\dot{H}_{per}$ and $\dot{V}_{per}$ respectively, and an orthonormal basis of H constituted by the eigenfunctions $(\mathbf{w}_i, i \in \mathbb{N})$ of the corresponding Stokes operator is given explicitly in Chapter 1, Section 1.5. Initial data and solutions can then be expanded in terms of these basis functions. For the initial data, we consider the expansion

$$\mathbf{u}_0(\omega; \mathbf{x}) = \sum_{i=1}^{\kappa} \sqrt{\lambda_i} Y_i(\omega) \mathbf{w}_i(\mathbf{x}) \, , \tag{7.1.1}$$

where $\kappa < \infty$ and Y_i are independent and *uniformly* distributed random variables on a bounded interval (a, b), and $\lambda_i = Ci^{-\gamma}$, with $\gamma > 1$. The initial probability distribution has then finite kinetic energy, as shown in Chapter 3, Section 3.2, i.e.

$$\int_H |\mathbf{v}|_H^2 \, d\mu_0(\mathbf{v}) = \|\mathbf{u}_0\|^2_{L^2(\Omega;H)} < \infty \, . \tag{7.1.2}$$

For the rest of this chapter, we will refer to the basis functions $\mathbf{w}_i(\mathbf{x})$ appearing in (7.1.1) as the functions $\mathbf{w}^{\mathcal{I}}_{\kappa_1,\kappa_2}(\mathbf{x}, t = 0)$, $\mathcal{I} \in \{I, II, III, IV\}$ described in Chapter 1, Section 1.5, normed with the coefficient $C_{\mathbf{w}} = 2$ such that they are of unit H-norm.

7.1.1 Generalized moment

We are interested in statistical moments, or ensemble averages of the flow, of the form

$$\mathbb{E}_{\mu_t}(\Phi) = \int_H \Phi(\mathbf{v}) \, d\mu_t(\mathbf{v}), \tag{7.1.3}$$

at some time $t > 0$.

For our numerical experiments, we will consider cylindrical test functions $\Phi \in \mathcal{C}$ of the form

$$\Phi(\mathbf{v}) = \phi\big((\mathbf{v}, \mathbf{g}_1)_H\big) \,, \tag{7.1.4}$$

that represent some bulk property of the flow, where ϕ is a compactly supported $\mathcal{C}^1$ function on $\mathbb{R}$ and $\mathbf{g}_1 \in V$. We choose here $\phi(x) = x$ (not formally compactly supported on $\mathbb{R}$, but it could be extended to a compactly supported function far away from the values of interest, assuming $(\mathbf{v}, \mathbf{g}_1)_H$ takes values on a bounded interval), such that

$$\Phi(\mathbf{v}) = (\mathbf{v}, \mathbf{g}_1)_H = \int_D \mathbf{v} \cdot \mathbf{g}_1 \, d\mathbf{x} \,. \tag{7.1.5}$$

The test function Φ satisfies the Lipschitz condition. Indeed, for $\mathbf{u}, \mathbf{v} \in H$, we have

$$\begin{aligned} |\Phi(\mathbf{u}) - \Phi(\mathbf{v})| &= |(\mathbf{u}, \mathbf{g}_1)_H - (\mathbf{v}, \mathbf{g}_1)_H| = |(\mathbf{u} - \mathbf{v}, \mathbf{g}_1)_H| \\ &\leq |\mathbf{u} - \mathbf{v}|_H^2 |\mathbf{g}_1|_H^2 = \underbrace{\big(|\mathbf{g}_1|_H^2 |\mathbf{u} - \mathbf{v}|_H\big)}_{C} |\mathbf{u} - \mathbf{v}|_H \,. \end{aligned} \tag{7.1.6}$$

Such a C always exists since $\mathbf{g}_1$ and $(\mathbf{u} - \mathbf{v})$ are in H.

7.1.2 MLMC estimator

We use the MLMC algorithm to calculate the statistical estimator $E^L_{\mu_t}(\Phi_L) \approx \mathbb{E}_{\mu_t}(\Phi)$ for the ensemble average of Φ at time $t > 0$, where $L > 0$ represents a (fine) discretization level. We have then a sequence of space and time discretization levels, where the discretization level $\ell = 0, \ldots, L$ is characterized by a meshwidth h_ℓ and a time step size $\Delta_\ell t$, and we denote by $\mathbf{u}_{\ell,\ell} = S_\ell(t, 0)\mathbf{v}$ the discrete solution of the Navier-Stokes equations with initial condition $\mathbf{v}$, where $S_\ell(t, 0)$ is the discrete solution operator that maps the initial data $\mathbf{v}$ into $\mathbf{u}_{\ell,\ell}$. In our simulations, we use the IMPACT solver described in Chapter 5 to calculate the discrete solutions $S_\ell(t, 0)\mathbf{v}$.

The MLMC estimator is, with $\Phi_\ell := \Phi(S_\ell(t, 0)\mathbf{v})$,

$$\begin{aligned} \mathbb{E}_{\mu_t}(\Phi) \approx E^L_{\mu_t}(\Phi_L) &= E^{M_0}_{\mu_t}(\Phi_0) + \sum_{\ell=1}^{L} E^{M_\ell}_{\mu_t}(\Phi_\ell - \Phi_{\ell-1}) \\ &= \frac{1}{M_0} \sum_{i=1}^{M_0} \Phi(S_0(t, 0)\mathbf{v}^i) + \sum_{\ell=1}^{L} \frac{1}{M_\ell} \sum_{i=1}^{M_\ell} (\Phi(S_\ell(t, 0)\mathbf{v}^i) - \Phi(S_{\ell-1}(t, 0)\mathbf{v}^i)) \,. \end{aligned} \tag{7.1.7}$$

With our choice of Φ and ϕ we have

$$\begin{aligned} E^L_{\mu_t}(\Phi_L) &= \frac{1}{M_0} \sum_{i=1}^{M_0} \big(S_0(t, 0)\mathbf{v}^i, \mathbf{g}_1\big)_H + \sum_{\ell=1}^{L} \frac{1}{M_\ell} \sum_{i=1}^{M_\ell} \big[\big(S_\ell(t, 0)\mathbf{v}^i, \mathbf{g}_1\big)_H - \big(S_{\ell-1}(t, 0)\mathbf{v}^i, \mathbf{g}_1\big)_H\big] \\ &= \big(E^{M_0}_{\mu_t}(\mathbf{u}_{0,0}), \mathbf{g}_1\big)_H + \sum_{\ell=1}^{L} \big[\big(E^{M_\ell}_{\mu_t}(\mathbf{u}_{\ell,\ell}), \mathbf{g}_1\big)_H - \big(E^{M_\ell}_{\mu_t}(\mathbf{u}_{\ell-1,\ell-1}), \mathbf{g}_1\big)_H\big] \\ &= \big(E^{M_0}_{\mu_t}(\mathbf{u}_{0,0}) + \sum_{\ell=1}^{L} \big[E^{M_\ell}_{\mu_t}(\mathbf{u}_{\ell,\ell}) - E^{M_\ell}_{\mu_t}(\mathbf{u}_{\ell-1,\ell-1})\big], \mathbf{g}_1\big)_H \\ &= \big(E^L_{\mu_t}(S_L(t, 0)\mathbf{u}_0), \mathbf{g}_1\big)_H \\ &= \Phi\big(E^L_{\mu_t}(S_L(t, 0)\mathbf{u}_0)\big) \,. \end{aligned} \tag{7.1.8}$$

That is, we can compute at first the MLMC estimate $E^L_{\mu_t}(S_L(t,0)\mathbf{u}_0)$ for the velocity field, and then apply the test function Φ, which is interesting if one wants to change the function $\mathbf{g}_1$ or the method used to evaluate $(\mathbf{u},\mathbf{g}_1)_H$. For test functions Φ for which we cannot do the above reasoning, we have to evaluate $\Phi^i_\ell \equiv \Phi(S_\ell(t,0)\mathbf{v}^i)$ after the sample $S_\ell(t,0)\mathbf{v}^i$ has been computed.

We remark that the addition of the vector fields $E^{M_\ell}_{\mu_t}(\mathbf{u}_{\ell,\ell})$ depends on the choice of the method used to reconstruct a solution in $L^2(D)$ from the discrete Finite Differences solutions outputted by the IMPACT solver. We prefer to have the liberty to choose the reconstruction method during the post-processing of results, therefore, instead of adding arrays of different sizes in our MLMC-FD solver (which would require to choose the reconstruction/interpolation method prior to the simulations), we output the arrays representing the discrete counterparts of $E^{M_\ell}_{\mu_t}(\mathbf{u}_{\ell,\ell})$ as such, and perform the reconstruction and addition needed to compute $E^L_{\mu_t}(S_L(t,0)\mathbf{u}_0)$ during the post-processing.

7.1.3 Space and time discretization

As already mentioned, we use the IMPACT solver described in Chapter 5 to compute the discrete solutions $S_\ell(t,0)\mathbf{v}$ for a given (random) initial velocity field $\mathbf{v}$. We discuss next the choices of h_ℓ and $\Delta_\ell t$ for the discretization on level ℓ, as well as the parameters used in the IMPACT code that are common to all the simulation results we present.

Since we have no solid walls that would need a refinement of the mesh size near the boundaries [10, chapt. 3.1.4], we partition the domain with an equidistant grid, and set the constant mesh size for level ℓ to

$$h_\ell = 2^{-\ell} \, . \tag{7.1.9}$$

The discretization scheme that we use in the IMPACT solver has, for sufficiently smooth solutions, a convergence order of at least $O(h^3_\ell)$ (the largest errors are on the boundaries [6, sect. 6.1.1]).

Time integration

For the time step size, we want to have a relation of the form $\Delta_\ell t = O(h_\ell)$, where the constant implied in the Landau symbol $O(\cdot)$ is independent of the viscosity ν. This is in general not possible with a fully explicit time integration scheme, because there the integration of the viscous term implies a viscosity-dependent restriction of the time step size, for stability reasons (see Chapter 5, Section 5.3). We use therefore the semi-implicit CN-RK3 integration scheme in IMPACT, which is of global order 2 for sufficiently smooth solutions.

For our case of periodic boundary conditions and no volume forces, there is no production of kinetic energy and the kinetic energy dissipates at a rate $\nu E(\mathbf{u})$, where $E(\mathbf{u})$ denotes the enstrophy (see Chapter 1, Section 1.4.1). We assume therefore that a time step size which is stable at the beginning of the time integration (with respect to the initial velocity field, at $t = 0$) remains stable for the whole simulation (i.e. also for $t > 0$). For our simulations we take a fixed time step size $\Delta_\ell t$, and for a given initial velocity sample $\mathbf{u}_0$, we use

$$\Delta_\ell t = \mathrm{CFL}(\mathbf{u}_0) h_\ell \, , \tag{7.1.10}$$

where the CFL-based parameter $\mathrm{CFL}(\mathbf{u}_0)$ depends on the maximal component-wise magnitude of the initial velocity field $\mathbf{u}_0$.

We set the threshold ε_H for the residual of the velocity field solution for the iterative solvers in IMPACT to be 10^{-10} (and allow a maximal number of 35 outer pressure iterations, see Chapter 5, Section 5.5), because it is approximately the accuracy required on the finest grids we use for the semi-implicit CN-RK3 to satisfy its expected convergence rate

of $O(\Delta_l t^2)$ (supposing enough smoothness of the solution). This accuracy is generally not necessary on coarser grids, but having it common to all simulations, together with constant meshwidths h_ℓ and constant time step sizes $\Delta_\ell t$, facilitates the pre- and post-simulation analysis.

We ensure zero-divergence of the discrete solutions by a flux correction vector that has minimal 2-norm (see Chapter 5, Section 5.5.6).

7.1.4 Error

We recall from Proposition 4.2.2 that if we can assume that, with $h_\ell \simeq \Delta_\ell t$, the space and time discretization error (in H-norm) is bounded by

$$|\mathbf{u}(t) - \mathbf{u}_{\ell,\ell}|_H \leq C \, \frac{h_\ell^\sigma}{\nu} \, , \tag{7.1.11}$$

for $\sigma > 0$ and with a constant $C > 0$ that is independent of ν, ℓ and h_ℓ, and that $h_{\ell-1} \leq \varrho h_\ell$, with $0 < \varrho < 1$, then, for a functional Φ satisfying the Lipschitz condition (4.2.2), the error bound

$$\|\mathbb{E}_{\mu_t}(\Phi) - E^L_{\mu_t}(\Phi_L)\|_{L^2(H;\mathbb{R})} \leq C(\varrho) \left(\frac{h_L^\sigma}{\nu} + \frac{1}{\sqrt{M_0}} + \sum_{\ell=0}^{L} \frac{1}{\sqrt{M_\ell}} \frac{h_\ell^\sigma}{\nu} \right) \tag{7.1.12}$$

holds, with a constant $C(\varrho)$ independent of L.

7.1.5 Error measurement

In our simulations, we will investigate if the error bound (7.1.12) holds. For this, we monitor the convergence of the error

$$\varepsilon_L^{\mathbb{E}} = |\mathbb{E}_{\mu_t}(\Phi)_{\text{ref}} - E^L_{\mu_t}(\Phi_L)| \tag{7.1.13}$$

in the $L^2(H,\mathbb{R})$-norm, where $\mathbb{E}_{\mu_t}(\Phi)_{\text{ref}}$ denotes the reference solution. Since the solution $E^L_{\mu_t}(\Phi_L)$ is a random variable, the discretization error $\varepsilon_L^{\mathbb{E}}$ is a random quantity as well. For the error convergence analysis we therefore compute a statistical estimator by averaging samples of $\varepsilon_L^{\mathbb{E}}$ from $K > 0$ independent runs and compute the error in (7.1.12) by approximating the $L^2(H,\mathbb{R})$-norm with Monte Carlo sampling, as explained in the following.

Let $\mathbb{E}_{\mu_t}(\Phi)_{\text{ref}}$ be the reference solution (i.e. the exact solution if it is known, otherwise an approximation of assumed higher accuracy as the MLMC estimates), and $(E^L_{\mu_t}(\Phi_L)^{(k)}, k = 1, ..., K)$ be a sequence of independent approximated solutions obtained by running the MLMC-FD solver K times.

Then the $L^2(H;\mathbb{R})$-based relative error estimator is defined [1, sect. 8.1] to be

$$\begin{aligned} \mathcal{R}\varepsilon_L^{\mathbb{E}} &:= \sqrt{\frac{1}{K}\sum_{k=1}^{K} \Big(\underbrace{\frac{\varepsilon_L^{\mathbb{E},(k)}}{|\mathbb{E}_{\mu_t}(\Phi)_{\text{ref}}|}}_{\mathcal{R}\varepsilon_L^{(k)}} \Big)^2} \times 100 \\ &\approx \| \underbrace{\frac{|\mathbb{E}_{\mu_t}(\Phi)_{\text{ref}} - E^L_{\mu_t}(\Phi_L)|}{|\mathbb{E}_{\mu_t}(\Phi)_{\text{ref}}|}}_{\mathcal{R}\varepsilon_L} \|_{L^2(H;\mathbb{R})} \times 100\% \, , \end{aligned} \tag{7.1.14}$$

where

$$\varepsilon_L^{\mathbb{E},(k)} = |\mathbb{E}_{\mu_t}(\Phi)_{\text{ref}} - E^L_{\mu_t}(\Phi_L)^{(k)}| \, . \tag{7.1.15}$$

If $\mathbb{E}_{\mu_t}(\Phi)_{\text{ref}} = 0$, we use

$$\mathcal{R}\varepsilon_L^{\mathbb{E}} := 100 \times \sqrt{\frac{1}{K}\sum_{k=1}^{K}\left(\varepsilon_L^{\mathbb{E},(k)}\right)^2} \tag{7.1.16}$$

instead.

In order to obtain a sufficiently accurate estimate of $\mathcal{R}\varepsilon_L$, the number K must be large enough to ensure a sufficiently small (< 0.01) [1, sect. 8.1] relative variance $\sigma^2(\mathcal{R}\varepsilon_L)$, which can be estimated [1, sect. 8.1] by

$$\sigma^2(\mathcal{R}\varepsilon_L) \approx \sigma_K^2(\mathcal{R}\varepsilon_L) := \frac{1}{K-1}\frac{E_{\mu_t}^K(\mathcal{R}\varepsilon_L^2 - E_{\mu_t}^K(\mathcal{R}\varepsilon_L)^2)}{E_{\mu_t}^K(\mathcal{R}\varepsilon_L)} \ . \tag{7.1.17}$$

We present next our results. The simulations were performed on Pilatus - Intel Sandy-Bridge (see Chapter 6, Section 6.2).

7.2 Numerical experiments

7.2.1 Discretization error in H-norm in the IMPACT code

First, we try to see if, for a smooth and laminar flow of the type of those that will be used in later experiments, an error bound in the H-norm of the form

$$|\mathbf{u} - \mathbf{u}_{\ell,\ell}|_H \leq C\frac{h^\sigma}{\nu} \tag{7.2.1}$$

holds, with a convergence rate $\sigma > 0$. This was an important assumption for the validity of the error bounds in Proposition 4.2.2.

We made two similar test cases with simple smooth and laminar flows. We can then assume smooth solutions, and we expect at least a second order convergence in h_ℓ, since the convergence of the spatial discretization scheme is at least of third order in h_ℓ and that of the time integration scheme of second order in $\Delta_\ell t$, and $\Delta_\ell t \simeq h_\ell$.

Test cases

The initial velocity field was chosen to be

$$\mathbf{u}_0(\mathbf{x}) = \frac{1}{2}\big(\mathbf{w}_{1,1}^I(\mathbf{x}) + \mathbf{w}_{1,1}^{II}(\mathbf{x}) + \mathbf{w}_{1,1}^{III}(\mathbf{x}) + \mathbf{w}_{1,1}^{IV}(\mathbf{x})\big) \ , \tag{7.2.2}$$

where the $\mathbf{w}_{1,1}^{\mathcal{I}}$ are the orthonormal eigenfunctions of the Stokes operator in the space-periodic case with vanishing space average. As shown in Chapter 1, Section 1.5, the exact solution at time t is

$$\mathbf{u}(\mathbf{x},t) = \mathbf{u}_0(\mathbf{x})e^{-4\pi^2(1^2+1^2)\nu t} \ . \tag{7.2.3}$$

In the first test, we have set the viscosity to $\nu = 0.01$ and the solution was calculated until $t = 0.1$, and in the second test, we have set the viscosity to $\nu = 0.1$ and the solution was calculated until $t = 0.01$.

The simulation was performed on a sequence of grids with mesh sizes $h_L = 2^{-L}$, $L = 4, \ldots, 10$. The constant time step size $\Delta_L t$ was set according to the component-wise magnitude of the initial velocity field. We have

$$\max_{\mathbf{x}\in D}|u_{0,1}(\mathbf{x})| \leq 4(\frac{1}{\sqrt{1^2+1^2}}) = \frac{4}{\sqrt{2}} \ , \tag{7.2.4}$$

$$\max_{\mathbf{x}\in D}|u_{0,2}(\mathbf{x})| \leq 4(\frac{1}{\sqrt{1^2+1^2}}) = \frac{4}{\sqrt{2}} \ . \tag{7.2.5}$$

With the CFL-condition

$$\Delta_L t \leq \mathrm{CFL}\frac{1}{\max\limits_{\mathbf{x}\in D}\{\frac{|u_{0,1}(\mathbf{x})|}{h_L}+\frac{|u_{0,2}(\mathbf{x})|}{h_L}\}} \tag{7.2.6}$$

in mind, we choose

$$\Delta_L t = 0.2\frac{h_L}{8/\sqrt{2}} = 0.025\sqrt{2}h_L \ . \tag{7.2.7}$$

We remark that for this choice of $\Delta_L t$, we calculate $\Delta_L t^2 = 1.28\times 2^{-10}h_L^2 \lesssim h_L^3$ for $L \leq 10$. Since the semi-implicit time integration scheme is of global order 2, we can expect the discretization error in time to be $O(\Delta_L t^2) = O(h_L^3)$ for $L \leq 10$. The spatial discretization scheme we use is of at least order 3, so we can expect in this case a spatio-temporal error of the order of $O(h_L^3)$ at the grid points. However, the discrete solution at time t is only given at those grid points, and we need to reconstruct it into a function in $L^2(D)$ in order to evaluate the bound (7.2.1). We denote this reconstruction by $\mathbf{u}_{\ell,\ell}^{\mathrm{rct}}(\mathbf{x})$.

We tested three methods to reconstruct the discrete solution: piecewise constant interpolation, bilinear interpolation and bicubic convolution interpolation [9]. All three are implemented in MATLAB. The bicubic convolution interpolation is suited for equidistant grids and has the advantage that it does not need to solve sub-problems (such as computing derivatives), which makes it efficient [9].

The error in (7.2.1) needs the evaluation of an integral

$$|\mathbf{u}-\mathbf{u}_{\ell,\ell}^{\mathrm{rct}}|_H^2 = \int_D \|\mathbf{u}-\mathbf{u}_{\ell,\ell}^{\mathrm{rct}}\|_2^2 \, d\mathbf{x} \ . \tag{7.2.8}$$

We used here a high-order composite 100-points 2D Gauss-Legendre quadrature to evaluate the integral, such that the integration error is negligible.

As expected, the convergence rate σ in (7.2.1) depends on the interpolation method used, as we can observe in Figure 7.1 for the first test case and in Figure 7.2 for the second test case. With piecewise constant interpolation, we have $\sigma = 1$, with bilinear interpolation, we have $\sigma = 2$, and with bicubic convolution interpolation, we have $\sigma = 3$.

In view of the sample number analysis from Chapter 4, Section 4.2.2, for our MLMC simulations, we will set the number of samples on each level to

$$M_\ell = M_L\big(\frac{h_\ell}{h_L}\big)^{2\sigma} = M_L 2^{2\sigma(L-\ell)}, \text{ for } \ell = 1,\dots,L, \quad \text{and} \quad M_0 = C_0\big(\frac{\nu}{h_L^\sigma}\big)^2 \ . \tag{7.2.9}$$

For $\sigma > 1$, this can result in quite large number of samples on the coarse levels, which would have taken a large amount of computing resources. Therefore we restricted ourselves to $\sigma = 1$, by choosing to evaluate $\Phi(\mathbf{u}_{\ell,\ell})$ using the piecewise constant reconstruction for $\mathbf{u}_{\ell,\ell}^{\mathrm{rct}}(\mathbf{x})$.

It may seem unnatural to have voluntary a larger discretization error, but this permits to test if the sample numbers used in the MLMC method and calculated for $\sigma = 1$ permit to equilibrate efficiently sampling and discretization errors on coarser levels such that the resulting final error is of the same order of magnitude as the discretization error on the finest level. Otherwise, one could argue that since we can expect third order convergence with an adequate interpolation method, the final error we observe is dominated by the sampling error and discretization errors are negligible in comparison, even on the coarser discretization levels. Also, in order to investigate the effect of under-resolved scales $h_\ell^\sigma > \nu$

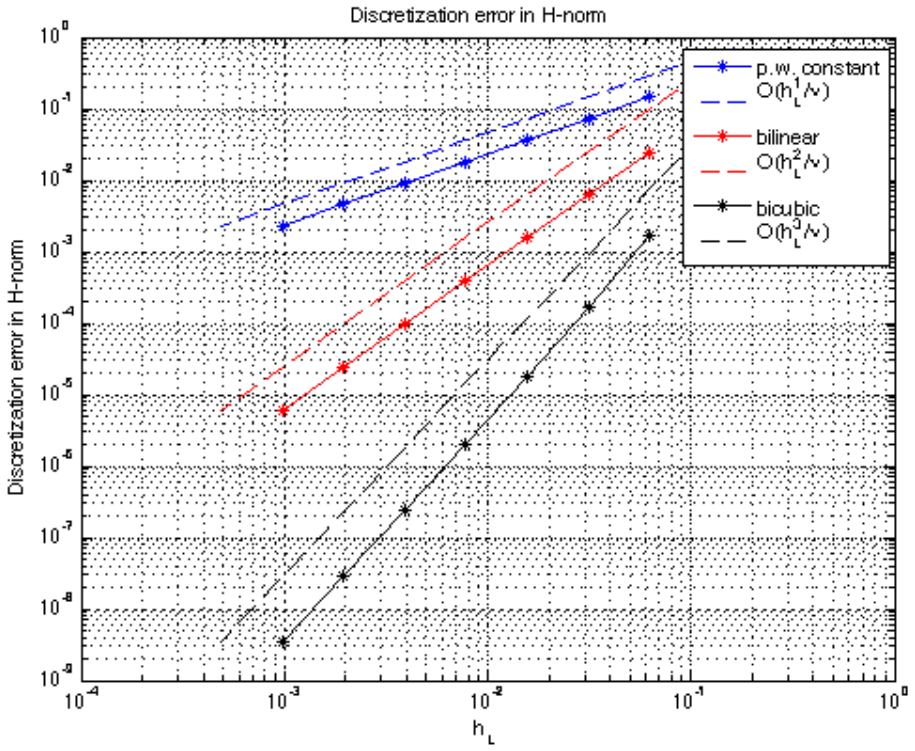


Figure 7.1: Test of the IMPACT code. Convergence of the error $|\mathbf{u} - \mathbf{u}_{L,L}^{\mathrm{rct}}|_H$ against the meshwidth h_L, for the case with $\nu = 0.01$ and $t = 0.1$. The FD solution has been interpolated on $D = (0,1) \times (0,1)$ with piecewise constant interpolation, bilinear interpolation and bicubic convolution interpolation (it is the bicubic interpolation MATLAB uses for equidistant grids), and integration to calculate the H-norm was performed with a composite 100-points 2D Gauss-Legendre quadrature rule. Figure generated with MATLAB.

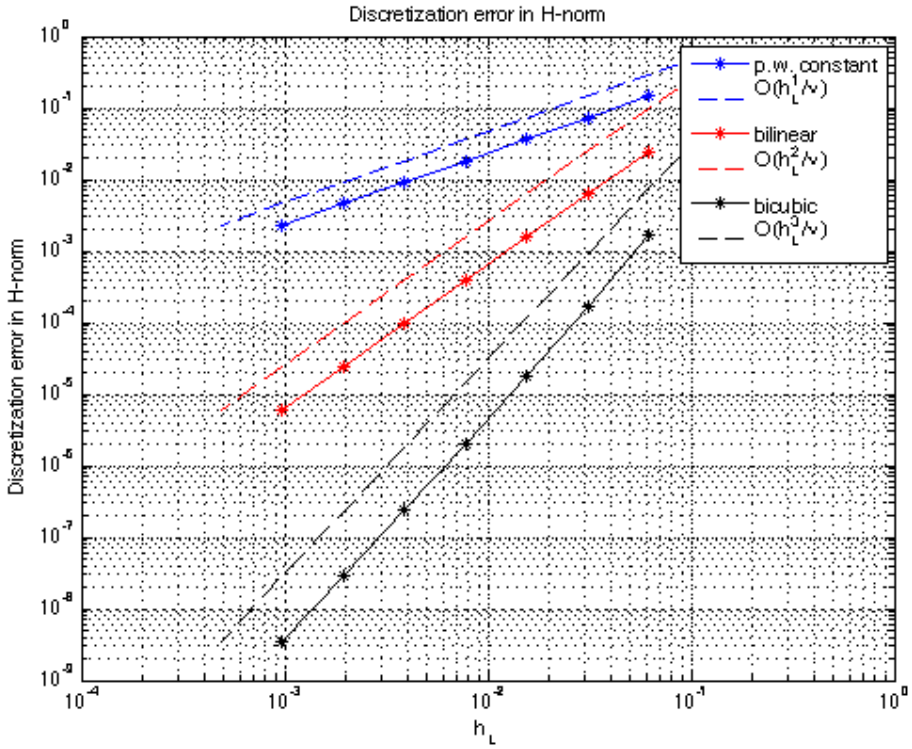


Figure 7.2: Test of the IMPACT code. Convergence of the error $|\mathbf{u} - \mathbf{u}_{L,L}^{\mathrm{rct}}|_H$ against the meshwidth h_L, for the case with $\nu = 0.1$ and $t = 0.01$. The FD solution has been interpolated on $D = (0,1) \times (0,1)$ with piecewise constant interpolation, bilinear interpolation and bicubic convolution interpolation (it is the bicubic interpolation MATLAB uses for equidistant grids), and integration to calculate the H-norm was performed with a composite 100-points 2D Gauss-Legendre quadrature rule. Figure generated with MATLAB.

in coarse levels ℓ, we would need, with $\sigma > 1$, smaller viscosity values ν as the one we use for our numerical experiments with smooth laminar flows.

We present next the results from our numerical experiments with the MLMC method, where we investigate the convergence of the error in (7.1.12), and we will see that, even in the presence of under-resolved discretization levels, the error bound holds.

7.2.2 MLMC - Test 1

Initial condition

In this test, the initial velocity was given by

$$\mathbf{u}_0(\omega;\mathbf{x}) = \sqrt{\lambda_1}Y_1(\omega)\big(\mathbf{w}^I_{1,1}(\mathbf{x}) + \mathbf{w}^{II}_{1,1}(\mathbf{x}) + \mathbf{w}^{III}_{1,1}(\mathbf{x}) + \mathbf{w}^{IV}_{1,1}(\mathbf{x})\big) \ , \tag{7.2.10}$$

where $\lambda_1 = \frac{1}{4}1^{-5}$ and $Y_1 \sim \mathcal{U}(0,1)$.
Using the orthonormality property of the Stokes eigenfunctions, we get

$$\begin{aligned} |\mathbf{u}_0(\omega)|^2_H &= (\mathbf{u}_0(\omega), \mathbf{u}_0(\omega))_H \\ &= \lambda_1 Y_1(\omega)^2 4 \\ &= Y_1(\omega)^2 \ . \end{aligned} \tag{7.2.11}$$

The mean kinetic energy of the initial condition is then

$$\mathbb{E}_{\mu_0}(|\mathbf{u}_0|^2_H) = \mathbb{E}_{\mu_0}(Y_1^2) = \mathrm{Var}(Y_1) + \mathbb{E}_{\mu_0}(Y_1)^2 = \frac{1}{3} \ . \tag{7.2.12}$$

Solution

As shown in Chapter 1, Section 1.5, the exact solution at time t is

$$\mathbf{u}(\omega;\mathbf{x},t) = \mathbf{u}_0(\omega;\mathbf{x})e^{-4\pi^2(1^2+1^2)\nu t} \ . \tag{7.2.13}$$

For our simulation, the end time was set to $t = 0.1$ and the viscosity to $\nu = 0.01$.

Time integration

The time step size choice depends on the random numbers appearing in the initial condition. We have

$$\max_{\mathbf{x}\in D}|u_{0,1}(\omega;\mathbf{x})| \leq 2\frac{1}{\sqrt{1^2+1^2}}|\sqrt{\lambda_1}Y_1(\omega)|4 \ , \tag{7.2.14}$$

$$\max_{\mathbf{x}\in D}|u_{0,2}(\omega;\mathbf{x})| \leq 2\frac{1}{\sqrt{1^2+1^2}}|\sqrt{\lambda_1}Y_1(\omega)|4 \ . \tag{7.2.15}$$

With the CFL-condition

$$\Delta_\ell t(\omega) \leq \mathrm{CFL}\frac{1}{\max\limits_{\mathbf{x}\in D}\{\frac{|u_{0,1}(\omega;\mathbf{x})|}{h_\ell} + \frac{|u_{0,2}(\omega;\mathbf{x})|}{h_\ell}\}} \tag{7.2.16}$$

in mind, we choose

$$\Delta_\ell t(\omega) = 0.2\frac{h_\ell}{u_{\max}(\omega)} \ , \tag{7.2.17}$$

where

$$u_{\max}(\omega) = \frac{2}{\sqrt{2}}\max\{1, 4|Y_1(\omega)|\} \tag{7.2.18}$$

is an upper bound for the velocity components. We use the maximum term to ensure that we do not have too large time step sizes.

MLMC

We investigate the convergence of the $L^2(H;\mathbb{R})$-error of the MLMC estimator $E^L_{\mu_t}(\Phi_L)$ by successfully increasing the finest level L from $L = 4$ to $L = 9$. For each L, the coarsest level contains $16 \times 16 = 256$ grid points. So $\ell = 0$ corresponds actually to the coarsest grid with meshwidth $h = 2^{-4}$, and we identify $\ell = 0$ with $\ell = 4$. In order to reduce the variance of the error estimates, we perform $K = 30$ independent simulation runs.

For the next results of the convergence of the error of the MLMC estimator presented in this thesis, we proceed similarly.

Sample numbers

The number of samples on the finest mesh is set to $M_L = 4$. As already mentioned, we have $\sigma = 1$, and we take the following sample numbers:

$$\begin{aligned} M_L &= 4\ , \\ M_\ell &= M_L\big(\frac{h_\ell}{h_L}\big)^2 = M_L 2^{2(L-\ell)}, \quad \ell = L-1, L-2, \dots, 1 \quad , \\ M_0 &= 100\big(\frac{\nu}{h_L^2}\big)^2 = 100\frac{\nu^2}{h_L^2}\ . \end{aligned} \tag{7.2.19}$$

We expect then that the error in Proposition 4.2.2 becomes

$$\|\mathbb{E}_{\mu_t}(\Phi) - E^L_{\mu_t}(\Phi_L)\|_{L^2(H;\mathbb{R})} \leq C\frac{h_L}{\nu}\ . \tag{7.2.20}$$

The term M_0, which is responsible for the purely sampling error, has a pre-factor of 100 to avoid too large relative errors. Recalling that $h_\ell = 2^{-\ell}$, we see that $h_\ell > \nu$ for $\ell \leq 6$, such that the discretization levels $\ell \leq 6$ do not satisfy the convergence requirement, i.e. they are under-resolved. Only the mesh resolution levels $\ell = 7, 8, 9$ are resolved.

We plot in Figure 7.3 and in Figure 7.4 the $L^2(H;\mathbb{R})$-based relative error $\mathcal{R}\varepsilon^{\mathbb{E}}_L$ in (7.1.14) against the meshwidth h_L on discretization level L for each of the following two functions $\mathbf{g}_1$ (recalling that $\Phi(\mathbf{v}) = (\mathbf{v}, \mathbf{g}_1)_H$),

$$\mathbf{g}_1(x_1, x_2) = \mathbf{w}^I(x_1, x_2) \tag{7.2.21}$$

and

$$\mathbf{g}_1(x_1, x_2) = (x_1 x_2, \frac{1}{2}x_2^2)^T\ . \tag{7.2.22}$$

The dashed lines indicate the expected convergence rate of the multilevel Monte Carlo method obtained in Proposition 4.2.2. This expected convergence rate coincides with the observations in the numerical experimental data, even in the presence of multiple under-resolved levels. The convergence in the resolved levels $\ell = 7, 8, 9$ indicates that the discretization error on the coarser samples is equilibrated by a high enough number of samples such that the total error is of the order of magnitude of $\frac{h_L}{\nu}$, even if the coarse samples are under-resolved.

In Figure 7.5 we can see the results corresponding to the only 5 first simulations with the function $\mathbf{g}_1$ in (7.2.21). The random nature of the error $\varepsilon^{\mathbb{E}}_L$ in (7.1.13) is there clearly visible, and it is therefore necessary to increase the number of simulations in order to reduce the variance of $\varepsilon^{\mathbb{E}}_L$ and obtain an accurate estimate of the $L^2(H;\mathbb{R})$-based relative error.

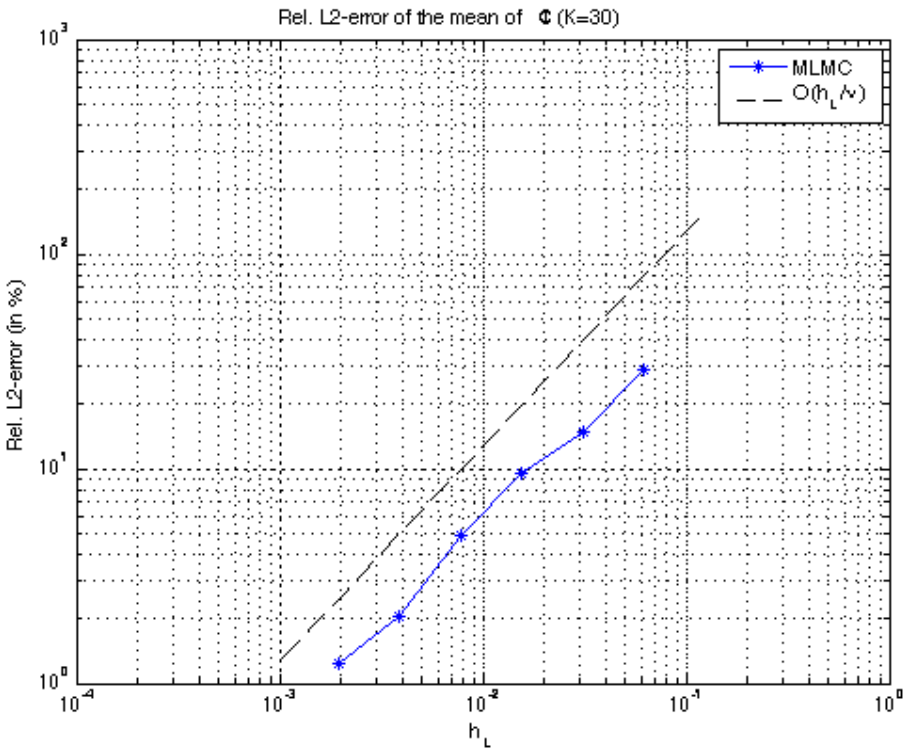


Figure 7.3: Test 1. Convergence of the relative error $\mathcal{R}\varepsilon_L^{\mathbb{E}}$ with $K = 30$ runs and $\mathbf{g}_1(x_1, x_2) = \mathbf{w}_{1,1}^I(x_1, x_2) \in V$. The reference solution was known. Piecewise constant reconstruction of the discrete solutions and composite 4-points 2D Gauss-Legendre quadrature were used for the evaluation of Φ. On all levels, the relative variance σ_K^2 (see 7.1.17) was at most 0.005. Figure generated with MATLAB.

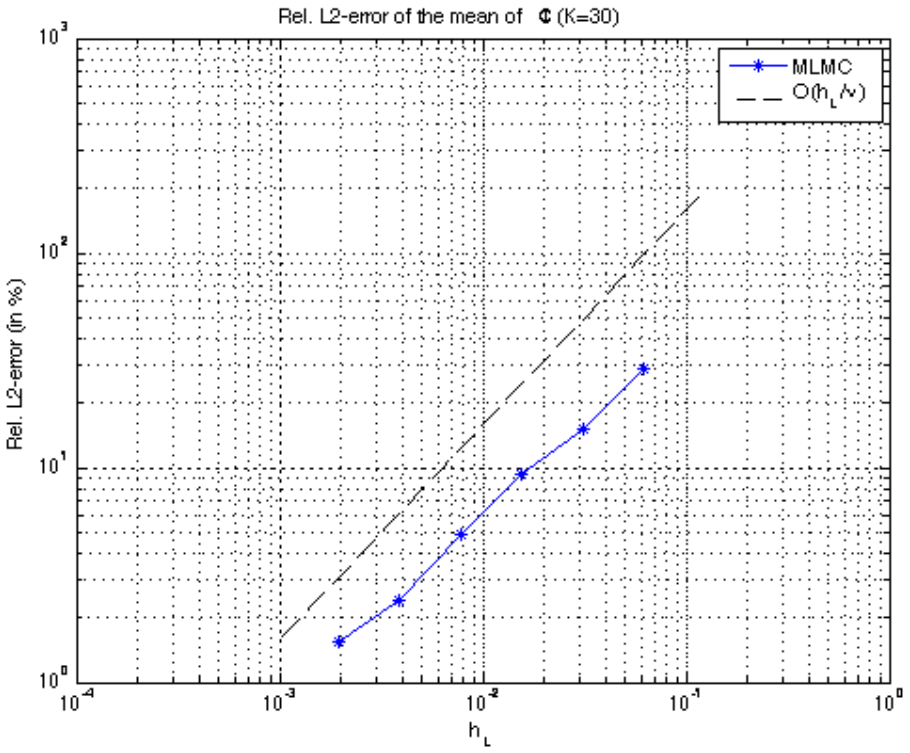


Figure 7.4: Test 1. Convergence of the relative error $\mathcal{R}\varepsilon_L^{\mathbb{E}}$ with $K = 30$ runs and $\mathbf{g}_1(x_1, x_2) = (x_1 x_2, \frac{1}{2}x_2^2)^T \in V$. The reference solution was calculated with 100-point 2D Gauss-Legendre quadrature. Piecewise constant reconstruction of the discrete solutions and composite 4-points 2D Gauss-Legendre quadrature were used for the evaluation of Φ. On all levels, the relative variance σ_K^2 was at most 0.005. Figure generated with MATLAB.

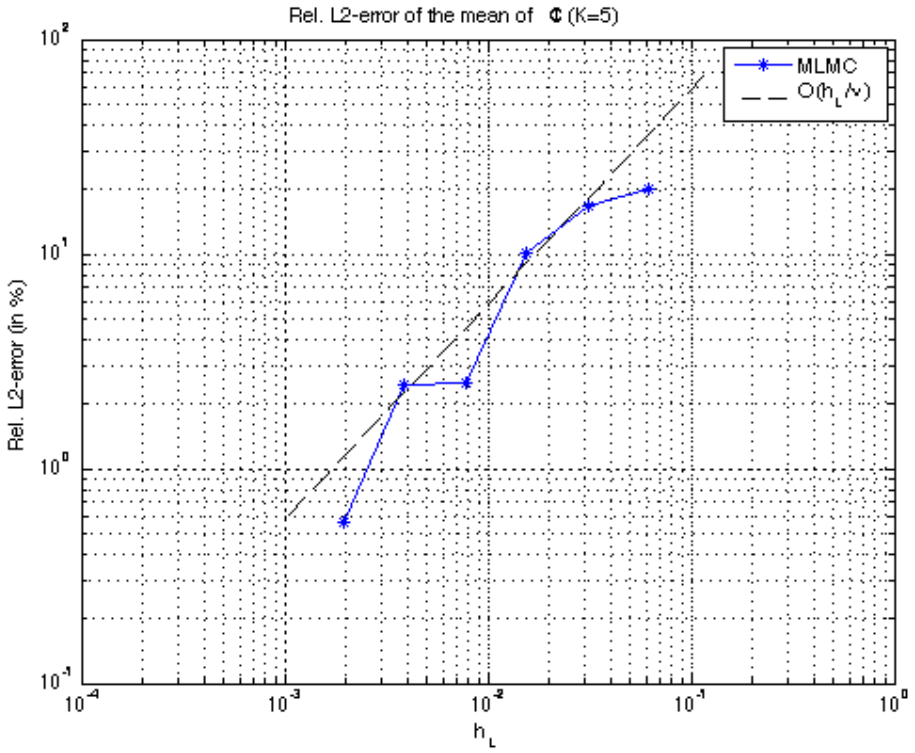


Figure 7.5: Test 1. Convergence of the relative error $\mathcal{R}\varepsilon_L^{\mathbb{E}}$ with $K = 5$ runs and $\mathbf{g}_1(x_1, x_2) = \mathbf{w}_{1,1}^I(x_1, x_2) \in V$. The reference solution was known. Piecewise constant reconstruction of the discrete solutions and composite 4-points 2D Gauss-Legendre quadrature were used for the evaluation of Φ. On all levels, the relative variance σ_K^2 was at most 0.029. Figure generated with MATLAB.

Parametrization

For the simulation, we used the processors assignment described in Table 7.1, where we use the notation introduced in Chapter 6. We report also in Table 7.2 the total number of cores used and the runtime (in seconds) needed to compute one estimate $E^L_{\mu_t}(\Phi_L)$ (averaged over $K = 30$ runs) measured by the MPI function `MPI_Wtime()`. For $L = 9$, we had to do 6 of the 30 runs separately because we had not allocated enough running time on the machine. The runtime of these 6 simulation runs are outliers compared to the other simulation runs, and we suspect a congestion of the network or a bad processor mapping to be the cause. Without these outliers, the (averaged) measured runtime on $L = 9$ is 489 (s).

Table 7.1: Test 1. Parametrization of $C_\ell = [D_\ell] \times P_\ell$.

L	C_4	C_5	C_6	C_7	C_8	C_9
4	$[1 \times 1] \times 1$	–	–	–	–	–
5	$[1 \times 1] \times 1$	$[2 \times 2] \times 1$	–	–	–	–
6	$[1 \times 1] \times 1$	$[2 \times 2] \times 1$	$[2 \times 2] \times 1$	–	–	–
7	$[1 \times 1] \times 2$	$[2 \times 2] \times 1$	$[2 \times 2] \times 1$	$[4 \times 4] \times 1$	–	–
8	$[1 \times 1] \times 4$	$[2 \times 2] \times 2$	$[2 \times 2] \times 1$	$[4 \times 4] \times 1$	$[4 \times 4] \times 1$	–
9	$[1 \times 1] \times 19$	$[2 \times 2] \times 8$	$[2 \times 2] \times 4$	$[4 \times 4] \times 2$	$[4 \times 4] \times 2$	$[4 \times 4] \times 2$

Table 7.2: Test 1. Total number of cores and runtime.

L	N_{cores}	Runtime (s)
4	1	1
5	5	2
6	9	9
7	26	68
8	48	163
9	163	1251

7.2.3 MLMC - Test 2

Initial condition

In this test, the initial velocity was given by

$$\mathbf{u}_0(\omega; \mathbf{x}) = \sqrt{\lambda_1} Y_1(\omega) \mathbf{w}^I_{1,1}(\mathbf{x}) + \sqrt{\lambda_2} Y_2(\omega) \mathbf{w}^I_{2,2}(\mathbf{x}) \,, \tag{7.2.23}$$

where $\lambda_i = \frac{1}{4} i^{-5}$ and $Y_i \overset{i.i.d.}{\sim} \mathcal{U}(0,1)$, $i = 1, 2$.
Using the orthonormality property of the Stokes eigenfunctions, we get

$$\begin{aligned} |\mathbf{u}_0(\omega)|^2_H &= (\mathbf{u}_0(\omega), \mathbf{u}_0(\omega))_H \\ &= \lambda_1 Y_1(\omega)^2 |\mathbf{w}^I_{1,1}|^2_H + \lambda_2 Y_2(\omega)^2 |\mathbf{w}^I_{2,2}|^2_H \\ &= \frac{1}{4}\big(Y_1(\omega)^2 + 2^{-5} Y_2(\omega)^2\big) \,. \end{aligned} \tag{7.2.24}$$

We calculate the mean kinetic energy of the initial condition with Monte Carlo sampling (10^6 samples), and get, with a 95%-confidence interval,

$$\mathbb{E}_{\mu_0}(|\mathbf{u}_0|_H^2) \approx 0.085995 \pm 0.000146 \, . \tag{7.2.25}$$

The end time was set to $t = 0.1$ and the viscosity to $\nu = 0.01$.

Reference solution

The reference solution is here unknown, and so we have to approximate it. To avoid method-related correlations, one should take a different method than MLMC to compute the reference solution. For example, one can take the MC method or a method based on numerical quadrature.

In this case, we use Gauss-Legendre quadrature to compute the reference solution $\mathbb{E}_{\mu_t}(\Phi)_{\text{ref}}$, together with discrete solutions $\mathbf{u}_{L,L}$ computed on the fine discretization level $L = 10$. We explain next the method.

Let us denote by $\Phi(\mathbf{u})^{(y_1,y_2)}$ the test function obtained when the initial condition $\mathbf{u}_0$ of $\mathbf{u}$ uses for its random coefficients the independent values $y_1 = Y_1(\omega)$ and $y_2 = Y_2(\omega)$ with probability distribution function $f_{Y1,Y2}$. Then we can write

$$\begin{aligned} E_{\mu_t}(\Phi(\mathbf{u})) &= \int_{(0,1)} \int_{(0,1)} \Phi(\mathbf{u})^{(y_1,y_2)} f_{Y_1,Y_2}(y_1,y_2) \, dy_1 \, dy_2 \\ &= \int_{(0,1)} \int_{(0,1)} \Phi(\mathbf{u})^{(y_1,y_2)} f_{Y_1}(y_1) f_{Y_2}(y_2) \, dy_1 \, dy_2 \, , \end{aligned} \tag{7.2.26}$$

because Y_1 and Y_2 are independent.
Since here $Y_i \overset{i.i.d.}{\sim} \mathcal{U}(0,1)$, $i = 1, 2$, we have $f_{Y_1}(y_1) = f_{Y_2}(y_2) = \mathbf{1}_{(0,1)}$, and we can write

$$\begin{aligned} E_{\mu_t}(\Phi(\mathbf{u})) &= \int_{(0,1)} \int_{(0,1)} \Phi(\mathbf{u})^{(y_1,y_2)} \, dy_1 \, dy_2 \\ &\approx \int_{(0,1)} \int_{(0,1)} \Phi(\mathbf{u}_{L,L})^{(y_1,y_2)} \, dy_1 \, dy_2 \\ &\approx \sum_{i=1}^{N} \sum_{j=1}^{N} \underbrace{w_{i,j}}_{=w_i w_j} \Phi(\mathbf{u}_{L,L})^{(\xi_i,\xi_j)} \, , \end{aligned} \tag{7.2.27}$$

where N is the number of quadrature points of the one-dimensional quadrature rule, and w_i, w_j and ξ_i, ξ_j are resp. the quadrature weights and the quadrature points on the interval $(0, 1)$. For the evaluation of the reference solution, we use here

$$\begin{aligned} \mathbb{E}_{\mu_t}(\Phi)_{\text{ref}} &= \sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j \Phi(\mathbf{u}_{L,L})^{(\xi_i,\xi_j)} \\ &= \Phi\big(\sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j \mathbf{u}_{L,L}^{(\xi_i,\xi_j)} \big) \\ &:= \Phi\big(\mathbf{u}_{ref}\big) \, , \end{aligned} \tag{7.2.28}$$

since our choice of Φ permits this exchange. We take here a quadrature rule with $N = 10$ quadrature points, such that the number of quadrature points on $(0, 1) \times (0, 1)$ is $N^2 = 100$. The advantage of calculating the reference solution with this method instead of with Monte Carlo is that the number of required discrete solutions, N^2, is smaller than M_L (we use

$N^2 = 100$ here instead of $M_L = \frac{\nu^2}{h_L^2} = 105$ for $\nu = 0.01$ and $h_L = 2^{-10}$), but also its assumed increased accuracy.

We may then wonder why we do not use Gaussian quadrature instead of (ML)MC sampling to approximate $\mathbb{E}_{\mu_t}(\Phi)$. For the small number of random coefficients used here for the initial condition (namely 2), it would be indeed more efficient, but as the complexity of the quadrature increases exponentially with the number of random coefficients, when we have 3 or more random coefficients, this method becomes not feasible anymore and we choose then (ML)MC, whose complexity increases only linearly with the number of random coefficients.

Time integration

The time step size choice depends on the random numbers appearing in the initial condition. We have

$$\begin{aligned}\max_{\mathbf{x}\in D}|u_{0,1}(\omega;\mathbf{x})| &\leq 2\frac{1}{\sqrt{1^2+1^2}}|\sqrt{\lambda_1}Y_1(\omega)| + 2\frac{2}{\sqrt{2^2+2^2}}|\sqrt{\lambda_2}Y_2(\omega)| \\ &= \frac{1}{\sqrt{2}}\big(|Y_1(\omega)| + |\sqrt{2^{-5}}Y_2(\omega)|\big)\ ,\end{aligned} \tag{7.2.29}$$

$$\begin{aligned}\max_{\mathbf{x}\in D}|u_{0,2}(\omega;\mathbf{x})| &\leq 2\frac{1}{\sqrt{1^2+1^2}}|\sqrt{\lambda_1}Y_1(\omega)| + 2\frac{2}{\sqrt{2^2+2^2}}|\sqrt{\lambda_2}Y_2(\omega)| \\ &= \frac{1}{\sqrt{2}}\big(|Y_1(\omega)| + |\sqrt{2^{-5}}Y_2(\omega)|\big)\ .\end{aligned} \tag{7.2.30}$$

With the CFL-condition

$$\Delta_\ell t(\omega) \leq \mathrm{CFL}\frac{1}{\max\limits_{\mathbf{x}\in D}\{\frac{|u_{0,1}(\omega;\mathbf{x})|}{h_\ell} + \frac{|u_{0,2}(\omega;\mathbf{x})|}{h_\ell}\}} \tag{7.2.31}$$

in mind, we choose

$$\Delta_\ell t(\omega) = 0.2\frac{h_\ell}{u_{\max}(\omega)}\ , \tag{7.2.32}$$

where

$$u_{\max}(\omega) = \frac{2}{\sqrt{2}}\max\{1, |Y_1(\omega)| + |\sqrt{2^{-5}}Y_2(\omega)|\} \tag{7.2.33}$$

is an upper bound for the velocity components. We use the maximum term to ensure that we do not have too large time step sizes.

MLMC

In this test, we considered an alternative, slightly stronger, requirement on the discretization error, i.e.

$$|\mathbf{u} - \mathbf{u}_{\ell,\ell}|_H \leq C\ \min\{1, \frac{h_\ell}{\nu}\}\ . \tag{7.2.34}$$

In this test also, the levels $\ell = 4, 5, 6$ are under-resolved. For the equilibration of statistical and discretization errors, the number of samples was set to

$$\begin{aligned} &M_L = 4\ , \\ &M_\ell = O\big((\frac{\min\{\nu, h_\ell\}}{h_L})^2\big) = \begin{cases} M_0, & h_\ell \geq \nu \\ M_L\big(\frac{h_\ell}{h_L}\big)^2, & h_\ell < \nu \end{cases}, \quad \ell = L-1, L-2, \ldots, 1 \quad , \\ &M_0 = O\big((\frac{\nu}{h_L})^2\big) = \begin{cases} M_L, & h_L \geq \nu \\ M_L\big(\frac{\nu}{h_L}\big)^2, & h_L < \nu \end{cases}\ . \end{aligned} \tag{7.2.35}$$

These special relations for the sample numbers are there to avoid having $M_L > M_\ell$ (with $\ell < L$) for the under-resolved finest levels $L = 4, 5, 6$, and also to avoid having sample numbers M_ℓ smaller than 1 for the under-resolved levels ℓ. Because of the relations in (7.2.35), the number of samples M_ℓ on the under-resolved levels is the same.

We expect that the error in Proposition 4.2.2 becomes

$$\|\mathbb{E}_{\mu_t}(\Phi) - E^L_{\mu_t}(\Phi_L)\|_{L^2(H;\mathbb{R})} \leq C \min\{1, \frac{h_L}{\nu}\} , \tag{7.2.36}$$

and we see in Figure 7.6 that we get this same error behavior, where we considered in the simulation the function $\mathbf{g}_1(x_1, x_2) = \mathbf{w}^I_{1,1}(x_1, x_2)$ for Φ. For the evaluation of $\Phi(\mathbf{u}_{ref}) = (\mathbf{u}_{ref}, \mathbf{g}_1)_H$, we used bicubic interpolation for the reconstruction and composite 25-points 2D Gauss-Legendre quadrature for the integration, such that the integration error is negligible.

We observe that for the under-resolved levels $\ell = 4, 5, 6$, whose discretization error is reduced to $O(1)$ by the relation (7.2.34), we recover this reduction of the error in the $L^2(H;\mathbb{R})$-based relative error. The convergence on higher levels $\ell = 7, 8, 9$ indicates that the larger discretization error on the coarser samples is equilibrated by a high enough number of samples such that the total error is of the order of magnitude of $\frac{h_L}{\nu}$, even if the coarse samples are under-resolved.

Parametrization

For the simulation, we used the processors assignment described in Table 7.3. We report also in Table 7.4 the total number of cores used and the runtime (in seconds) needed to compute one estimate $E^L_{\mu_t}(\Phi_L)$ (averaged over $K = 30$ runs) measured by the MPI function `MPI_Wtime()`.

Table 7.3: Test 2. Parametrization of $C_\ell = [D_\ell] \times P_\ell$.

L	C_4	$C_{4\ \&\ 5}$	C_6	C_7	C_8	C_9
4	$[1 \times 1] \times 1$	–	–	–	–	–
5	–	$[1 \times 1] \times 1$	–	–	–	–
6	–	$[1 \times 1] \times 1$	$[2 \times 2] \times 1$	–	–	–
7	–	$[1 \times 1] \times 1$	$[2 \times 2] \times 1$	$[4 \times 4] \times 1$	–	–
8	–	$[1 \times 1] \times 1$	$[2 \times 2] \times 1$	$[4 \times 4] \times 1$	$[4 \times 4] \times 2$	–
9	–	$[1 \times 1] \times 1$	$[2 \times 2] \times 1$	$[4 \times 4] \times 1$	$[4 \times 4] \times 2$	$[8 \times 8] \times 1$

Table 7.4: Test 2. Total number of cores and runtime.

L	N_{cores}	Runtime (s)
4	1	1
5	1	1
6	5	3
7	21	15
8	53	66
9	117	265

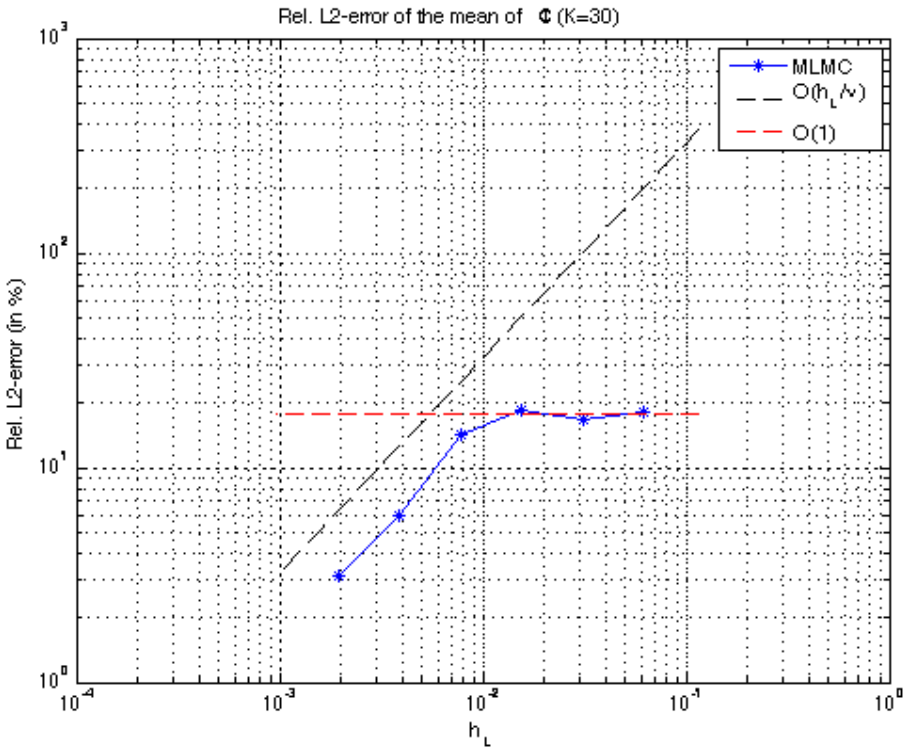


Figure 7.6: Test 2. Convergence of the relative error $\mathcal{R}\varepsilon_L^{\mathbb{E}}$ with $K = 30$ runs and $\mathbf{g}_1(x_1, x_2) = \mathbf{w}_{1,1}^I(x_1, x_2) \in V$. The reference solution was calculated with 100-points 2D Gauss-Legendre quadrature. Piecewise constant reconstruction of the discrete solutions and composite 4-points 2D Gauss-Legendre quadrature were used for the evaluation of Φ. On all levels, the relative variance σ_K^2 was at most 0.0032. Figure generated with MATLAB.

7.2.4 MLMC - Test 3

Initial condition

In this test, the initial velocity was given by

$$\mathbf{u}_0(\omega;\mathbf{x}) = \sqrt{\lambda_1}Y_1(\omega)\mathbf{w}^I_{1,1}(\mathbf{x}) + \sqrt{\lambda_2}Y_2(\omega)\mathbf{w}^I_{2,2}(\mathbf{x}) \ , \tag{7.2.37}$$

where $\lambda_i = \frac{1}{4}i^{-5}$ and $Y_i \overset{i.i.d.}{\sim} \mathcal{U}(-1,1)$, $i = 1,2$.
Using the orthonormality property of the Stokes eigenfunctions, we get

$$\begin{aligned} |\mathbf{u}_0(\omega)|^2_H &= (\mathbf{u}_0(\omega),\mathbf{u}_0(\omega))_H \\ &= \lambda_1 Y_1(\omega)^2|\mathbf{w}^I_{1,1}|^2_H + \lambda_2 Y_2(\omega)^2|\mathbf{w}^I_{2,2}|^2_H \\ &= \frac{1}{4}\big(Y_1(\omega)^2 + 2^{-5}Y_2(\omega)^2\big) \ . \end{aligned} \tag{7.2.38}$$

We calculate the mean kinetic energy of the initial condition with Monte Carlo sampling (10^6 samples), and get, with a 95%-confidence interval,

$$\mathbb{E}_{\mu_0}(|\mathbf{u}_0|^2_H) \approx 0.60178 \pm 0.00043 \ . \tag{7.2.39}$$

The end time was set to $t = 0.01$ and the viscosity to $\nu = 0.1$.

Time integration

The time step size choice depends on the random numbers appearing in the initial condition. We have

$$\begin{aligned} \max_{\mathbf{x}\in D}|u_{0,1}(\omega;\mathbf{x})| &\leq 2\frac{1}{\sqrt{1^2+1^2}}|\sqrt{\lambda_1}Y_1(\omega)| + 2\frac{2}{\sqrt{2^2+2^2}}|\sqrt{\lambda_2}Y_2(\omega)| \\ &= \frac{1}{\sqrt{2}}\big(|Y_1(\omega)| + |\sqrt{2^{-5}}Y_2(\omega)|\big) \ , \end{aligned} \tag{7.2.40}$$

$$\begin{aligned} \max_{\mathbf{x}\in D}|u_{0,2}(\omega;\mathbf{x})| &\leq 2\frac{1}{\sqrt{1^2+1^2}}|\sqrt{\lambda_1}Y_1(\omega)| + 2\frac{2}{\sqrt{2^2+2^2}}|\sqrt{\lambda_2}Y_2(\omega)| \\ &= \frac{1}{\sqrt{2}}\big(|Y_1(\omega)| + |\sqrt{2^{-5}}Y_2(\omega)|\big) \ . \end{aligned} \tag{7.2.41}$$

With the CFL-condition

$$\Delta_\ell t(\omega) \leq \mathrm{CFL}\frac{1}{\max\limits_{\mathbf{x}\in D}\{\frac{|u_{0,1}(\omega;\mathbf{x})|}{h_\ell} + \frac{|u_{0,2}(\omega;\mathbf{x})|}{h_\ell}\}} \tag{7.2.42}$$

in mind, we choose

$$\Delta_\ell t(\omega) = 0.2\frac{h_\ell}{u_{\max}(\omega)} \ , \tag{7.2.43}$$

where

$$u_{\max}(\omega) = \frac{2}{\sqrt{2}}\max\{1, |Y_1(\omega)| + |\sqrt{2^{-5}}Y_2(\omega)|\} \tag{7.2.44}$$

is an upper bound for the velocity components. We use the maximum term to ensure that we do not have too large time step sizes.

Reference solution

The reference solution was calculated here with the Monte Carlo method on the discretization level $L = 10$. The number of samples was chosen according to $M_L = \left(\frac{\nu}{h_L}\right)^2 \approx 10486$. We took $M_L = 10010$ for our simulation to limit the required computing resources.

A unique multilevel Monte Carlo estimate on the discretization level $L = 10$ has also been calculated, with the same sample numbers choice as in (7.2.45), in order to compare approximatively the computational cost needed by both methods to attain the same accuracy.

MLMC

All levels $\ell = 4, \dots, 9$ are here resolved, and for the MLMC simulation, we took the following sample numbers:

$$\begin{aligned} M_L &= 4 \ , \\ M_\ell &= M_L \big(\frac{h_\ell}{h_L}\big)^2 = M_L 2^{2(L-\ell)}, \quad \ell = L-1, L-2, \dots, 1 \quad , \\ M_0 &= \big(\frac{\nu}{h_L^2}\big)^2 = \frac{\nu^2}{h_L^2} \ , \end{aligned} \tag{7.2.45}$$

and we expect then that the error in Proposition 4.2.2 becomes

$$\|\mathbb{E}_{\mu_t}(\Phi) - E^L_{\mu_t}(\Phi_L)\|_{L^2(H;\mathbb{R})} \leq C \frac{h_L}{\nu} \ . \tag{7.2.46}$$

In the simulation, we considered the function $\mathbf{g}_1(x_1, x_2) = \mathbf{w}^I_{1,1}(x_1, x_2)$ for Φ. The reference solution was reconstructed with bicubic interpolation and the evaluation of Φ for the reference solution was done with composite 25-points 2D Gaussian quadrature, such that the integration error is negligible. The behavior predicted in (7.2.46) can be observed in Figure 7.7.

Parametrization

For the simulation, we used the processors assignment described in Table 7.5. We report also in Table 7.6 the total number of cores used and the runtime (in seconds) needed to compute one estimate $E^L_{\mu_t}(\Phi_L)$ (averaged over $K = 30$ runs, except for the MLMC simulation on level $L = 10$, where we report the timing of the only run we made) measured by the MPI function `MPI_Wtime()`.

Table 7.5: Test 3. Parametrization of $C_\ell = [D_\ell] \times P_\ell$.

L	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
4	$[1 \times 1] \times 1$	–	–	–	–	–	–
5	$[1 \times 1] \times 1$	$[2 \times 2] \times 1$	–	–	–	–	–
6	$[1 \times 1] \times 1$	$[2 \times 2] \times 1$	$[2 \times 2] \times 1$	–	–	–	–
7	$[1 \times 1] \times 2$	$[2 \times 2] \times 1$	$[2 \times 2] \times 1$	$[4 \times 4] \times 1$	–	–	–
8	$[1 \times 1] \times 4$	$[2 \times 2] \times 2$	$[2 \times 2] \times 1$	$[4 \times 4] \times 1$	$[4 \times 4] \times 1$	–	–
9	$[1 \times 1] \times 19$	$[2 \times 2] \times 8$	$[2 \times 2] \times 4$	$[4 \times 4] \times 2$	$[4 \times 4] \times 2$	$[4 \times 4] \times 2$	–
10	$[1 \times 1] \times 14$	$[2 \times 2] \times 8$	$[2 \times 2] \times 4$	$[4 \times 4] \times 2$	$[4 \times 4] \times 2$	$[4 \times 4] \times 4$	$[16 \times 16] \times 1$

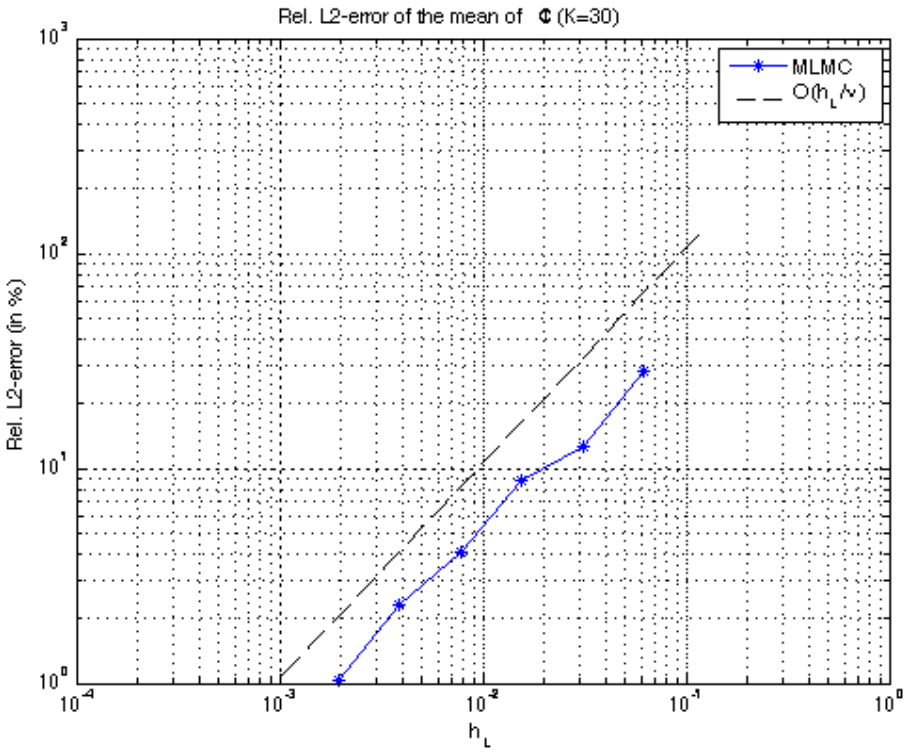


Figure 7.7: Test 3. Convergence of the relative error $\mathcal{R}\varepsilon_L^{\mathbb{E}}$ with $K = 30$ runs and $\mathbf{g}_1(x_1, x_2) = \mathbf{w}_{1,1}^I(x_1, x_2) \in V$. The reference solution was calculated with the Monte Carlo method with 10010 samples on the discretization level $L = 10$. Piecewise constant reconstruction of the discrete solutions and composite 4-points 2D Gauss-Legendre quadrature were used for the evaluation of Φ. On all levels, the relative variance σ_K^2 was at most 0.0035. Figure generated with MATLAB.

Table 7.6: Test 3. Total number of cores and runtime.

L	N_{cores}	Runtime (s)
4	1	1
5	5	1
6	9	3
7	26	14
8	48	41
9	163	130
10	446	311

Comparison of MLMC and MC

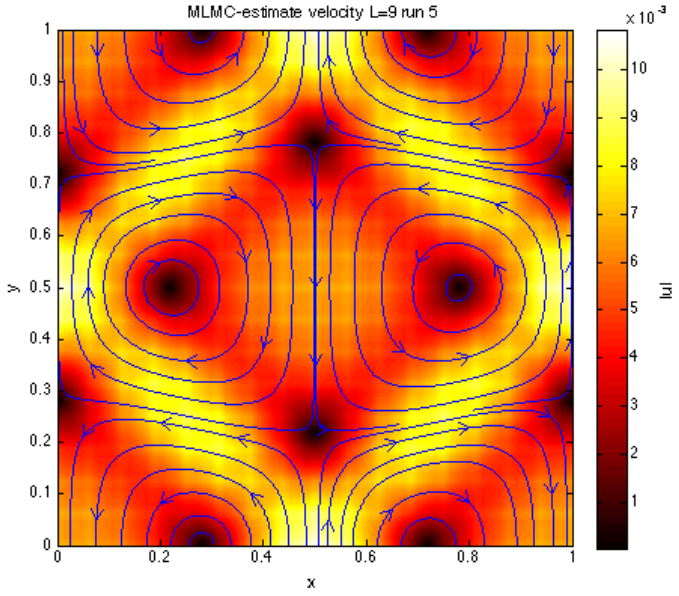
As mentioned previously, for this simulation we made both a Monte Carlo simulation on the fine discretization level $L = 10$, and a multilevel Monte Carlo simulation with finest discretization level $L = 10$. In both cases, the sample numbers were chosen in order to obtain a $L^2(H;\mathbb{R})$-based relative error of the order of magnitude of $\frac{h_L}{\nu}$. The parametrization $C_{10} = D_{10} \times P_{10}$ for the processors assignment in the Monte Carlo simulation was $C_{10} = [8 \times 8] \times 14 = 896$ cores. The parametrization and the number of cores used in the multilevel Monte Carlo simulation on the finest level $L = 10$ can be seen in Table 7.5 and Table 7.6.

We can get a crude estimation of the total computational work by multiplying on each level the number of processors assigned on that level with the (average) time needed for the time integration in that level. This takes also into account the communication time in the IMPACT solver, for this reason it is only an estimation.

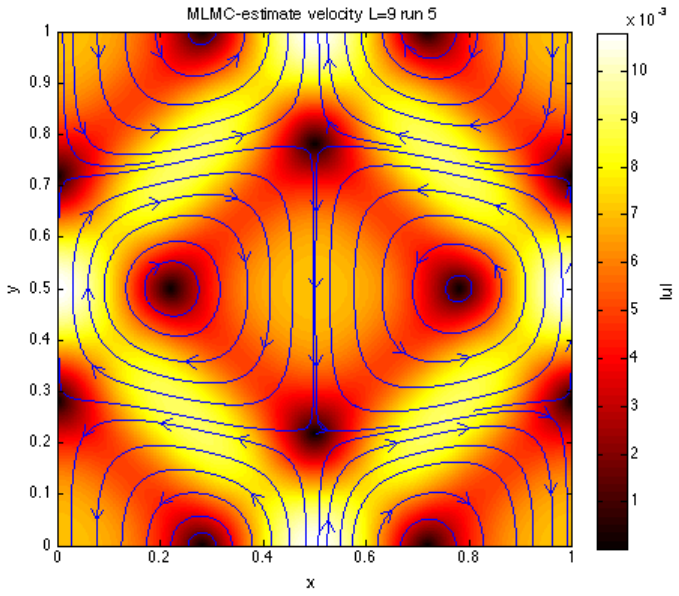
We measured the ratio of the computational work to calculate the MC estimate over the computational work to calculate the MLMC estimate to be about 690, such that the MLMC simulation had a cost of about two orders of magnitude smaller.

Effect of the reconstruction method on the velocity field

To conclude the discussion about this test, we present an example resulting from the MLMC simulation to illustrate the fact that, while the piecewise constant reconstruction seems to be sufficiently accurate for the evaluation of the real-valued functional Φ (representing some bulk property of the flow), one may loose intrinsic smoothness of the FD solution with this method, such that for the reconstruction e.g. of the mean velocity field, one should use a higher order interpolation scheme. Indeed, interpolation schemes such as piecewise constant interpolation or bilinear interpolation may display artifacts, for example when adding flows rotating in different directions (we remark that the random coefficients in the initial condition may also take negative values in this test). This is visible by comparing Figure 7.8a, where bilinear interpolation was used to reconstruct the MLMC estimate for the mean velocity field in one of the simulation runs on level $L = 9$, and Figure 7.8b, where bicubic interpolation was used for the same simulation run on that level.



(a) Test 3. MLMC estimate $E^L_{\mu_t}(S_L(t,0)\mathbf{u}_0)$ on level $L = 9$ for the simulation run number 5. Bilinear reconstruction of the discrete solutions was performed. Figure generated with MATLAB.



(b) Test 3. MLMC estimate $E^L_{\mu_t}(S_L(t,0)\mathbf{u}_0)$ on level $L = 9$ for the simulation run number 5. Bicubic reconstruction of the discrete solutions was performed. Figure generated with MATLAB.

Figure 7.8: Effect of the reconstruction method on the velocity field.

Efficiency

While efficiency is of great importance in parallel applications, for the numerical experiments, we were primarily focused on calculating correctly the statistical estimates and on observing the expected error convergence rates, while trying naturally to guess *a priori*, from prior results, the best processor assignment to achieve an efficient load balancing.

Final words

The presented results have shown that the MLMC-FD method seems to capture correctly ensemble averages and bulk properties of the flow, even in the presence of under-resolved discretizations, and confirm thus the theoretical foundings presented in Chapter 4.

Acknowledgments

We thank the team of CSCS [13] for providing support and computational resources under the project ID 'g54'.

Conclusion

The MLMC-FD solver implemented in the context of this thesis approximates ensemble averages and bulk properties of statistical solutions of the Navier-Stokes equations by sampling, combined with the use of the discrete solver IMPACT for the pathwise evolution of each sample arising out of an ensemble of initial conditions. It is based on the novel theoretical and computational approach presented in [1] and the implementation approach (in a parallel environment) presented in [17].

The simulation results showed that the errors arising from coarse, under-resolved discretizations used in the MLMC algorithm can be compensated (in mean square sense) efficiently by statistical oversampling, such that the resulting approximations of the ensemble averages attain the level of accuracy of the finest discretizations used in the algorithm, and this at a lower cost than with the more traditional MC approach.

Outlook

For our simulations, we considered smooth and laminar flows, in two space dimensions and with periodic boundary conditions. Since the discussion leading to the error bounds for the MLMC estimates in Chapter 4 took also into account no-slip boundary conditions and three space dimensions, we expect to obtain similar convergence results as the one observed in our numerical experiments for this type of boundary conditions and for this higher space dimension, assuming smooth and laminar flows.

The error bounds for the MLMC estimates were derived assuming robust convergence (with respect to the grid spacing) of the discretization error measured in the energy-norm. Since this norm can be seen (when squared) as a spatial average of the discretization error in the (squared) euclidean norm, the discrete solutions do not need to resolve locally and in detail the small scale features of the flow, as long as this spatial average converges consistently, i.e. as long as these discrete solutions resolve the bulk properties of the flow independently of the small scale features of the flow.

At larger Reynolds numbers however, even with this less restrictive character of the energy-norm, this is only possible with the use of a proper turbulence model on under-resolved simulations. The investigation of the effect of using turbulence models on the accuracy and the efficiency of the MLMC algorithm could be an idea for future work.

Bibliography

[1] A. Barth, Ch. Schwab, and J. Šukys. Multilevel Monte Carlo approximations of statistical solutions of the Navier-Stokes equations. SAM report 2013-33, November 2013.

[2] C. Foias, O. Manley, R. Rosa, and R. Temam. *Navier-Stokes Equations and Turbulence*, volume 83 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1st edition, 2001.

[3] C. Foias, R. Rosa, and R. Temam. Properties of time-dependent statistical solutions of the three-dimensional Navier-Stokes equations. *Annales de l'Institute Fourier (to appear)*, 2013. ArXiV 1111.6257v2.

[4] R. Henniger. *Direct and large-eddy simulation of particle transport processes in estuarine environments*. PhD thesis, ETH Zurich, 2011.

[5] R. Henniger. IMPACT simulation code. `http://www.ifd.mavt.ethz.ch/research/group_lk/projects/impact`, August 2011.

[6] R. Henniger, D. Obrist, and L. Kleiser. High-order accurate solution of the incompressible Navier-Stokes equations on massively parallel computers. *Journal of Computational Physics*, 229:3543,3572, 2010.

[7] R. Hiptmair, Ch. Schwab, H. Harbrecht, V. Gradinaru, A. Chernov, and P. Grohs. Numerical Methods for Partial Differential Equations. `http://www.sam.math.ethz.ch/~hiptmair/tmp/NPDE12.pdf`, March 2012. Lecture notes.

[8] K.-I. Ishikawa. Multiple stream Mersenne Twister PRNG. `http://theo.phys.sci.hiroshima-u.ac.jp/~ishikawa/PRNG/mt_stream_en.html`, March 2011.

[9] R. G. Keys. Cubic Convolution Interpolation for Digital Image Processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153,1160, December 1981.

[10] L. Kleiser. Berechnungsmethoden der Energie- und Verfahrenstechnik. `http://www.ifd.mavt.ethz.ch/education/Lectures/`, Februar 2012. Lecture notes.

[11] L. Kleiser and T. Rösgen. Fluiddynamik I/II. `http://www.ifd.mavt.ethz.ch/education/Lectures/`, November 2011. Lecture notes.

[12] P. K. Kundu and I. M. Cohen. *Fluid Mechanics*. Academic Press, 4th edition, 2008.

[13] Swiss National Supercomputing Center (CSCS). Pilatus - Intel SandyBridge. `www.cscs.ch`.

[14] S. B. Pope. *Turbulent Flows*. Cambridge University Press, 9th edition, 2011.

[15] The Portland Group. `http://www.pgroup.com/resources/mvapich/mvapich_2011.htm`.

[16] B. Rummler. *Zur Lösung der instationären inkompressiblen Navier-Stokesschen Gleichungen in speziellen Gebieten*. PhD thesis, Otto-von-Guericke-Universität Magdeburg, May 2000. Anhang A.1 Die Eigenfunktionen im Perioden-Quader.

[17] J. Šukys, S. Mishra, and Ch. Schwab. Static Load Balancing for Multi-level Monte Carlo Finite Volume Solvers. In R. Wyrzykowski et al., editor, *Parallel Processing and Applied Mathematics*, volume 7203 of *Lecture Notes in Computer Science*, pages 245,254. Springer Berlin Heidelberg, 2012.