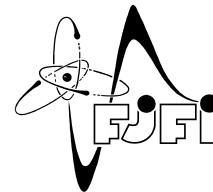




CZECH TECHNICAL UNIVERSITY IN PRAGUE  
Faculty of Nuclear Sciences and Physical Engineering



# **Simulation of turbulent flow and contaminant dispersion in a street network**

## **Simulace turbulentního proudění přenosu znečišťujících příměsí v uliční síti**

Master's Thesis

Author: **Adam Gottfried**  
Supervisor: **Mgr. Vladimír Fuka, Ph.D.**  
Consultant: **Ing. Pavel Strachota, Ph.D.**  
Academic year: 2024/2025

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Gottfried** Jméno: **Adam** Osobní číslo: **494657**  
Fakulta/ústav: **Fakulta jaderná a fyzikálně inženýrská**  
Zadávací katedra/ústav: **Katedra matematiky**  
Studijní program: **Matematické inženýrství**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Simulace turbulentního proudění přenosu znečišťujících příměsí v uliční síti**

Název diplomové práce anglicky:

**Simulation of turbulent flow and contaminant dispersion in a street network**

Pokyny pro vypracování:

1. Navrhněte vhodnou kombinaci numerických schémat a okrajových podmínek v metodě konečných prvků se spektrálními/hp elementy pro řešení nestlačitelného proudění v numerickém kódu Nektar++.
2. Proveďte simulace proudění v idealizované městské zástavbě s liniovým zdrojem znečištění.
3. Porovnejte výsledky výpočtů s výsledky měření a s existujícími výpočty metodou konečných diferencí.
4. Nerozlišená turbulence je v Nektar++ modelována jako implicitní simulace velkých vírů (implicit large eddy simulation). Porovnejte výsledná spektra turbulentních fluktuací s teoretickými předpoklady, s výsledky měření a s výsledky existujících simulací s explicitními modely nerozlišené turbulence.

Seznam doporučené literatury:

- [1] C.D. Cantwell et al., Nektar++: An open-source spectral/hp element framework, Computer Physics Communications 192, 2015, 205-219.
- [2] W. Hamblí, J. Slaughter, F. F. Buscariolo, S. Sherwin, Extension of Spectral/hp Element Methods towards Robust Large-Eddy Simulation of Industrial Automotive Geometries. Fluids 7(3), 2022, 106.
- [3] Š. Nosek, V. Fuka, L. Kukačka, Z. Kluková, Z. Jaňour, Street-canyon pollution with respect to urban-array complexity: The role of lateral and mean pollution fluxes, Building and Environment 138, 2018, 221-234

Jméno a pracoviště vedoucí(ho) diplomové práce:

**Mgr. Vladimír Fuka, Ph. D. MFF UK**

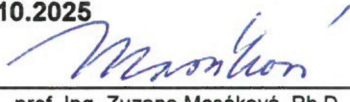
Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:


**Ing. Pavel Strachota, Ph.D. katedra matematiky FJFI**

Datum zadání diplomové práce: **31.10.2023** Termín odevzdání diplomové práce: **10.05.2024**

Platnost zadání diplomové práce: **31.10.2025**

  
Mgr. Vladimír Fuka, Ph. D.  
podpis vedoucí(ho) práce

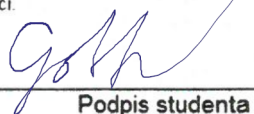
  
prof. Ing. Zuzana Masáková, Ph.D.  
podpis vedoucí(ho) ústavu/katedry

  
doc. Ing. Václav Čuba, Ph.D.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

10.11.2023  
Datum převzetí zadání

  
Podpis studenta



*Acknowledgment:*

I would like to thank Mgr. Vladimír Fuka, Ph.D. for his expert supervision and express my gratitude to Ing. Pavel Strachota, Ph.D. for his time and advice.

*Author's declaration:*

I declare that this Master's Thesis is entirely my own work and I have listed all the used sources in the bibliography.

Prague, August 4, 2025

Adam Gottfried

*Název práce:*

**Simulace turbulentního proudění přenosu znečišťujících příměsí v uliční síti**

*Autor:* Adam Gottfried

*Studijní program:* Matematické inženýrství

*Druh práce:* Diplomová práce

*Vedoucí práce:* Mgr. Vladimír Fuka, Ph.D., MFF UK

*Konzultant:* Ing. Pavel Strachota, Ph.D., katedra matematiky FJFI

*Abstrakt:* Efektivní metody pro výpočet turbulentního proudění jsou již dlouho předmětem intenzivního výzkumu. Takovou metodou je implicitní metoda velkých vírů pomocí spektrální mizející viskozity, jíž se zabývá tato práce. Byla navržena geometrie spolu s okrajovými podmínkami vhodnými pro modelování transportu pasivního skaláru v síti uličních kaňonů. Na této geometrii byly vypočteny simulace proudění metodou spektrálních/hp prvků za užití spektrální mizející viskozity, která je implementována v prostředí Nektar++. Výsledná pole, včetně spekter turbulentní energie, byla porovnána s ostatními metodami pro modelování turbulence a s experimentálními daty.

*Klíčová slova:* metoda spektrálních prvků, simulace velkých vírů, spektrální mizející viskozita, uliční kaňon

*Title:*

**Simulation of turbulent flow and contaminant dispersion in a street network**

*Author:* Adam Gottfried

*Abstract:* Effective ways to compute turbulent flow fields have long been a subject of intense research. One such method, the implicit large eddy simulation using spectral vanishing viscosity is investigated in this work. Suitable geometry and boundary conditions are proposed to model turbulent transport of passive scalar in a street canyon network. In this geometry, simulations are run using the spectral/hp element method implemented with spectral vanishing viscosity in Nektar++ framework. The results including energy spectra are studied and compared to other methods of modeling turbulent flows and with experimental data.

*Key words:* large eddy simulation, spectral element method, spectral vanishing viscosity, street canyon

# Contents

<b>Introduction</b>	<b>7</b>
<b>1 Turbulent flows</b>	<b>8</b>
1.1 Turbulence spectra . . . . .	9
1.2 Kolmogorov hypotheses . . . . .	11
1.3 Large-eddy simulation . . . . .	12
1.4 Spectral vanishing viscosity . . . . .	15
1.5 Passive scalar advection . . . . .	16
<b>2 Spectral/hp element method</b>	<b>17</b>
2.1 Spectral method . . . . .	17
2.2 Finite element method . . . . .	18
2.3 Spectral/hp element method . . . . .	19
2.4 SVV for the spectral/hp element method . . . . .	21
2.5 Gauss-Lobatto-Legendre basis . . . . .	22
2.6 Quasi 3D simulation . . . . .	23
<b>3 Numerical results</b>	<b>26</b>
3.1 Problem set-up . . . . .	26
3.2 Software tools used for the simulations . . . . .	28
3.3 Comparison between iLES and DNS . . . . .	29
3.4 Passive scalar advection with line source . . . . .	32
3.5 Passive scalar advection with point source . . . . .	33
<b>Conclusion</b>	<b>36</b>
<b>Appendix</b>	<b>37</b>

# Introduction

Simulating turbulent flows has long been an intensely studied topic, with many applications ranging from meteorology [40] to automotive engineering [2]. One such field where turbulent flows are of interest is the simulation of pollution spread in urban street networks [7][28], which will be the focus of this work. Since pollutant transport differs between laminar and turbulent flows, a reliable approach to simulating turbulent flow is required to compute the transport with any desired accuracy.[31]

Turbulent flows have long posed a great challenge to physicists and mathematicians due to their chaotic nature [23] and the scales at which important processes take place [21][20][31]. It soon became clear that the Navier-Stokes equations, used to describe incompressible flows, were not the ideal way to approach turbulence for many applications. Thus, different models of turbulence, such as Reynolds averaging [32] and large eddy simulations [33], were developed. Reynolds-averaged models approach turbulence as a random field and compute the average flows, while large eddy simulations attempt to filter out smaller scales of turbulent motion and preserve the larger scales.

We will focus on large eddy simulations, which have taken many forms over the years and found wide use, particularly in atmospheric sciences [13][28][34] and engineering [12]. Among them, a specific type of implicit large eddy simulation called spectral vanishing viscosity was developed. This approach uses viscous regularization that only affects the high spectral modes of the turbulent flow [35][18].

The models of turbulence are closely tied to the numerical methods used to solve the incompressible flow equations. Aside from finite differences [14] and finite volumes [26], spectral methods [6] and finite element methods [41] have become popular for this type of simulation. Combining the last two methods, a new approach called the spectral element method has been developed [29][19]. This method takes advantage of the geometric versatility of finite element methods while preserving some of the advantageous convergence properties of spectral methods. This method, together with the spectral method, is especially suitable for implementing the spectral vanishing viscosity model of turbulence.

The goal of this work is to find a suitable geometry and set of boundary conditions to model the turbulent transport of pollutants in a street canyon. Then, we perform simulations on this geometry using the spectral element method with the spectral vanishing viscosity turbulence model. The resulting turbulent flow is compared to other methods of simulating turbulent flows, especially in terms of energy spectra, and the pollutant transport will be compared to experimental wind tunnel measurements.

This work is divided into three chapters. The first chapter deals with different methods for modeling turbulent flows, especially large eddy simulations and spectral vanishing viscosity. The second chapter summarizes the various numerical methods used in turbulent flow simulation and places them in the context of the aforementioned models of turbulence. The third chapter discusses the results of our simulation and compares them to other approaches, as well as to experimental data.

# Chapter 1

## Turbulent flows

This chapter introduces the methods used to deal with turbulent flows. Most of its content follows [31], which provides further detail. We will focus only on the parts necessary for our simulations.

As mentioned in the introduction, we focus on the incompressible flow of a Newtonian fluid in thermodynamic equilibrium. Such flow is described by the set of Navier-Stokes equations [4]

$$\frac{D\mathbf{U}}{Dt} = \nu \nabla^2 \mathbf{U} - \frac{1}{\rho} \nabla P, \quad (1.1)$$

$$\nabla \cdot \mathbf{U} = 0. \quad (1.2)$$

Although this set of equations fully describes the problem, such a description is not satisfactory for studying turbulent flows, because important processes such as energy dissipation happen at small scales and the macroscopic flows exhibit chaotic behavior. To address this issue, we treat the velocity field  $\mathbf{U}(t, \mathbf{x})$  as a stochastic variable, parameterized by time and space coordinates.

Due to the chaotic nature of the Navier-Stokes equations (as shown most notably in [23]), it is often useful to work with mean values of the flow fields. For that reason, one might formulate equations describing the dynamics of these statistics. We use the Reynolds decomposition, which separates the random field into its mean (denoted  $\langle \mathbf{U} \rangle$ ) and the fluctuation from that mean (denoted  $\mathbf{u}$ )

$$\mathbf{U}(t, \mathbf{x}) = \langle \mathbf{U} \rangle(t, \mathbf{x}) + \mathbf{u}(t, \mathbf{x}). \quad (1.3)$$

Taking the mean of the material derivative, we obtain

$$\left\langle \frac{DU_i}{Dt} \right\rangle = \frac{\partial \langle U_i \rangle}{\partial t} + \frac{\partial \langle U_i U_j \rangle}{\partial x_j} \quad (1.4)$$

$$= \frac{\partial \langle U_i \rangle}{\partial t} + \langle U_j \rangle \frac{\partial \langle U_i \rangle}{\partial x_j} + \frac{\partial \langle u_i u_j \rangle}{\partial x_j}. \quad (1.5)$$

Since the other terms in equations (1.1) and (1.2) are linear, we can simply interchange the mean value and the derivative when taking the mean of the whole equation. This results in the mean momentum equation and the mean divergence-free equation

$$\frac{\langle D \rangle \langle \mathbf{U} \rangle}{\langle D \rangle t} = \nu \nabla^2 \langle \mathbf{U} \rangle - \frac{1}{\rho} \nabla \langle P \rangle - \nabla \cdot \langle \mathbf{u} \mathbf{u}^T \rangle, \quad (1.6)$$

$$\nabla \cdot \langle \mathbf{U} \rangle = 0, \quad (1.7)$$

where  $\frac{\langle D \rangle}{\langle D \rangle t}$  is the material derivative along the mean flow, and  $\langle P \rangle$  is the mean pressure. We can see that this set of equations differs from the original one by the term on the right-hand side involving the



covariance matrix of the fluctuations. Note that with the introduction of the fluctuating terms, the set of equations is no longer complete and raises the need for a model of turbulence.

Rewriting the first equation as

$$\rho \frac{\langle D \rangle \langle U_i \rangle}{\langle D \rangle t} = \frac{\partial}{\partial x_i} \left[ \mu \left( \frac{\partial \langle U_j \rangle}{\partial x_i} - \frac{\partial \langle U_i \rangle}{\partial x_j} \right) + \langle P \rangle \delta_{ij} - \rho \langle u_i u_j \rangle \right], \quad (1.8)$$

we can see that the covariance matrix can be interpreted as a stress term together with the viscous and diagonal isotropic stress. For this reason, the term  $\langle u_i u_j \rangle$  is referred to as Reynolds stress.

**Remark.** *Although the full stress term is  $-\rho \langle u_i u_j \rangle$ , convention dictates that the term Reynolds stress refers only to the covariance of the fluctuation, stripped of the minus sign and density.*

Reynolds stresses form a second-order symmetric positive semi-definite tensor. To define turbulent kinetic energy, we take half of the trace of this tensor, i.e.,

$$k = \frac{1}{2} \langle \mathbf{u} \cdot \mathbf{u} \rangle. \quad (1.9)$$

This value represents the mean kinetic energy of a unit mass in the fluctuating flow field. The Reynolds stress tensor can be further decomposed into its isotropic and anisotropic parts, respectively, as

$$\langle u_i u_j \rangle = \frac{2}{3} k \delta_{ij} + a_{ij}, \quad (1.10)$$

where the isotropic part can be included in the modified pressure term, as can be seen from (1.8).

The results above lead us to our first model of turbulence: the turbulence-viscosity hypothesis [31]. This model assumes that the deviatoric parts of the stress terms are proportional to the mean viscous stress term, i.e.,

$$-\rho \langle u_i u_j \rangle + \frac{2}{3} k \rho \delta_{ij} = \rho \nu_T \left( \frac{\partial \langle U_j \rangle}{\partial x_i} - \frac{\partial \langle U_i \rangle}{\partial x_j} \right). \quad (1.11)$$

This allows us to rewrite equation (1.8) using the effective viscosity  $\nu_{eff} = \nu + \nu_T$  as

$$\frac{\langle D \rangle \langle U_i \rangle}{\langle D \rangle t} = \frac{\partial}{\partial x_i} \left[ \nu_{eff} \left( \frac{\partial \langle U_j \rangle}{\partial x_i} - \frac{\partial \langle U_i \rangle}{\partial x_j} \right) - \frac{1}{\rho} (\langle P \rangle + \frac{2}{3} \rho k) \delta_{ij} \right], \quad (1.12)$$

restoring completeness to our system of equations, provided we have a good description of  $\nu_{eff}$ . However, it turns out that the assumption that the anisotropic Reynolds stress tensor  $a_{ij}$  is aligned with the mean rate-of-strain tensor  $\langle S_{ij} \rangle$  is too strong to accurately model turbulent flows [36]. Therefore, we must explore more sophisticated ways to model turbulence.

## 1.1 Turbulence spectra

The Reynolds stresses are, as mentioned above, covariances between the components of the fluctuation at the same point in space and at the same time. One might also define the covariances for different times and different points in space. Let us first assume a fixed point  $\mathbf{x}$  and a statistically stationary flow. We then define the temporal autocovariance (without Einstein summation) as

$$R_i(s, \mathbf{x}) := \langle u_i(t, \mathbf{x}) u_i(t + s, \mathbf{x}) \rangle. \quad (1.13)$$

The formula above is to be understood without Einstein summation on the right-hand side. If we take its Fourier transform, we get the frequency spectrum of the turbulence (in the  $i$ -th component), which is a quantity that will be important for our analysis of the turbulent flows.

$$E^i(\omega, \mathbf{x}) := \frac{1}{\pi} \int_{-\infty}^{\infty} R_i(s, \mathbf{x}) e^{-i\omega s} ds. \quad (1.14)$$

Similarly, we might take the covariance of the fluctuation between two points at the same time. Assuming a statistically homogeneous flow, we define

$$R_i(t, \mathbf{r}) := \langle u_i(t, \mathbf{x}) u_i(t, \mathbf{x} + \mathbf{r}) \rangle, \quad (1.15)$$

again without Einstein summation, and its Fourier transform as

$$\phi_i(t, \boldsymbol{\kappa}) := \frac{1}{(2\pi)^3} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-i\boldsymbol{\kappa} \cdot \mathbf{r}} R_i(t, \mathbf{r}) d\mathbf{r}. \quad (1.16)$$

The quantity  $\phi_i(t, \boldsymbol{\kappa})$  represents the contribution of the harmonic modes of the wavelength  $|\boldsymbol{\kappa}|$  in the direction  $\boldsymbol{\kappa}$  to (the covariance of) the fluctuation  $u_i$ . We can also define a quantity that represents this contribution across all directions. This quantity is called the energy spectrum, and it is defined as

$$E^x(t, \kappa) := \sum_{i=1}^D \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{2} \phi_i(t, \boldsymbol{\kappa}) \delta(|\boldsymbol{\kappa}| - \kappa) d\boldsymbol{\kappa}. \quad (1.17)$$

The reason for the name of the quantity, as well as the one-half coefficient, becomes clear when we look at its relation to the turbulent kinetic energy

$$k = \frac{1}{2} \langle \mathbf{u} \cdot \mathbf{u} \rangle = \frac{1}{2} \sum_{i=1}^D R_i(t, 0) = \sum_{i=1}^D \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{2} \phi_i(t, \boldsymbol{\kappa}) d\boldsymbol{\kappa} \quad (1.18)$$

$$= \int_0^{\infty} E(t, \kappa) d\kappa. \quad (1.19)$$

From this, we can see that  $E(t, \kappa)$  represents the contribution of modes with wavenumber  $\kappa$  to the turbulent kinetic energy.

A careful reader has noticed that we have used the same notation for both the temporal and the spatial covariance. This will not pose any problems, as we will mostly be dealing with flows that are both statistically homogeneous (in  $x_i$ ) and statistically stationary. In these types of flows, Taylor's hypothesis applies, which states that the covariances can be interchanged through the following identity if  $|\mathbf{u}| \ll \langle \mathbf{U} \rangle$  [37]

$$R_i(s, \mathbf{x}) = \langle u_i(t, \mathbf{x}) u_i(t + s, \mathbf{x}) \rangle \approx \langle u_i(t, \mathbf{x}) u_i(t, \mathbf{x} + s\langle \mathbf{U} \rangle) \rangle = R_i(t, \mathbf{x} + s\langle \mathbf{U} \rangle). \quad (1.20)$$

For practical reasons, we will want to also consider one-dimensional energy spectra, which take into account only those wavenumbers that are aligned with one of the coordinate axes. Therefore, we define

$$E_i(t, \kappa_1) = \frac{1}{\pi} \int_{-\infty}^{\infty} R_i(t, r_1 \mathbf{e}_1) e^{-ir_1 \kappa_1} dr_1 = \frac{2}{\pi} \int_0^{\infty} R_i(t, r_1 \mathbf{e}_1) \cos(r_1 \kappa_1) dr_1. \quad (1.21)$$

The inverse transformation gives

$$R_i(t, r_1 \mathbf{e}_1) = \int_0^{\infty} E_i(t, \kappa_1) \cos(r_1 \kappa_1) d\kappa_1. \quad (1.22)$$

For  $r_1 = 0$ , we get

$$\langle u_i^2 \rangle = R_i(t, 0) = \int_0^\infty E_i(t, \kappa_1) d\kappa_1. \quad (1.23)$$

Comparing this result to (1.18), we can see that knowing the one-dimensional spectra for each component of the velocity is sufficient for computing the turbulent kinetic energy.

## 1.2 Kolmogorov hypotheses

One of the characteristic properties of turbulence is its self-similarity [16][39]. This means that large vortices (eddies) are broken up into smaller versions of themselves, and so on. At each scale, one can characterize the eddies by their characteristic length, time, and velocity. Large eddies transfer their kinetic energy to the smaller ones. This process continues until the eddies are small enough that viscous dissipation absorbs the kinetic energy of the small eddies. An important implication of this notion is that the kinetic energy dissipates only in the smallest eddies. The rate of kinetic energy dissipation in the system,  $\varepsilon$ , is then determined by the behavior at the smallest scales and the rate of energy transfer between the scales.

This qualitative view of turbulence needs to be complemented by a quantitative one, which answers questions about the characteristic dimensions of the smallest eddies as well as the rate of energy dissipation. These answers are provided by the Kolmogorov hypotheses [21][20]. They gave rise to a theory of turbulence that divides the energy cascade described above into three regions: the energy-containing range of the largest eddies, the inertial subrange of the eddies transferring energy from the top to the bottom of the cascade, and the dissipative range of the smallest eddies. For us, the most important hypotheses can be stated as follows [31]

1. In every turbulent flow at a sufficiently high Reynolds number, the statistics of the motions in the inertial subrange and the dissipative range have a universal form that is uniquely determined by  $\nu$  and  $\varepsilon$ .
2. In every turbulent flow at a sufficiently high Reynolds number, the statistics of the motions in the inertial subrange have a universal form that is uniquely determined by  $\varepsilon$ , independent of  $\nu$ .

These hypotheses have important implications. The first hypothesis answers the question about the characteristic length, velocity, and time at the smallest scale. These are, respectively,

$$\eta = (\nu^3/\varepsilon)^{\frac{1}{4}}, \quad (1.24)$$

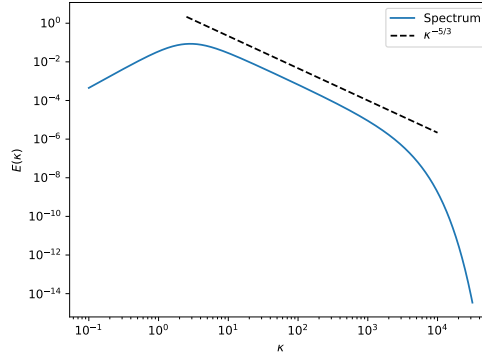
$$u_\eta = (\nu\varepsilon)^{\frac{1}{4}}, \quad (1.25)$$

$$\tau_\eta = (\nu/\varepsilon)^{\frac{1}{2}}. \quad (1.26)$$

These correspond to the Reynolds number of the smallest eddies being one. The second hypothesis can then be shown to imply that the rate of energy transfer between scales is constant in the inertial subrange, i.e., independent of scale.

As a consequence of the first Kolmogorov hypothesis, the energy spectra have a universal form [31], which can be written as

$$E(\kappa) = (\varepsilon\nu^5)^{\frac{1}{4}}\psi(\eta\kappa). \quad (1.27)$$

Figure 1.1: Modeled spectrum for  $\eta = 0.0001$  and  $L = 1$ .

Alternatively, this can be expressed as

$$E(\kappa) = \varepsilon^{\frac{2}{3}} \kappa^{-\frac{5}{3}} \Psi(\eta\kappa), \quad (1.28)$$

where

$$\Psi(\eta\kappa) = (\eta\kappa)^{\frac{5}{3}} \psi(\eta\kappa). \quad (1.29)$$

Finally, the second hypothesis tells us that for the inertial subrange,  $E(\kappa)$  is independent of  $\nu$ , which enters the equation (1.28) solely through  $\eta$ . This means that  $\Psi$  in the inertial subrange must be independent of its argument, i.e., a constant. This gives us the power law for the spectrum

$$E(\kappa) = C \varepsilon^{\frac{2}{3}} \kappa^{-\frac{5}{3}}. \quad (1.30)$$

The same argument can be applied to the one-dimensional spectra.

To illustrate the behavior of the whole spectrum, we can look at a model [22][38][31], where

$$\Psi(\eta\kappa) = e^{-\beta\eta\kappa} \left( \frac{\kappa L}{\sqrt{(\kappa L)^2 + c_L}} \right)^{\frac{11}{3}}. \quad (1.31)$$

The full spectrum can then be written as

$$E(\kappa) = C \varepsilon^{\frac{2}{3}} \kappa^{-\frac{5}{3}} e^{-\beta\eta\kappa} \left( \frac{\kappa L}{\sqrt{(\kappa L)^2 + c_L}} \right)^{\frac{11}{3}}. \quad (1.32)$$

Here,  $C, \beta, c_L$  are parameters usually chosen as  $C = 1.5, \beta = 5.2$  and  $c_L = 6.78$ .  $L$  is the characteristic length of the largest eddies. Large eddy behavior is ensured by the term in the brackets. This term tends to one for large  $\kappa$ . The exponential term then represents the dissipative range and tends to one as  $\kappa$  decreases. With a suitably chosen set of parameters, there is a clearly visible inertial range, where both terms in  $\Psi$  are close to one and the spectrum behavior is dictated by  $\kappa^{-\frac{5}{3}}$ . This is illustrated in Figure 1.1.

### 1.3 Large-eddy simulation

One of the methods that has been developed to model and compute turbulent flows is the large-eddy simulation (LES) [33]. As the name suggests, this method tries to resolve the large eddies numerically

and substitute the smaller scales with a suitable model. The idea behind this is that, as the Kolmogorov hypotheses state, the small-scale motions have a universal form, and their effects should therefore be possible to model. The central operation in LES is filtering, which is used to filter out the small-scale motions from the field  $\varphi$  by convoluting it with some filter  $G(x)$ , creating a new filtered flow field  $\bar{\varphi}$ :

$$\bar{\varphi}(\mathbf{x}) = \int_{\Omega} G(\mathbf{r})\varphi(\mathbf{x} - \mathbf{r}) \, d\mathbf{r}. \quad (1.33)$$

For the filtered velocity field, a similar process to Reynolds averaging can be done, separating the original field into its filtered and residual components

$$\mathbf{U} = \bar{\mathbf{U}} + \mathbf{u}'. \quad (1.34)$$

An important difference from Reynolds averaging is that the filtered velocity is still a random variable. Using the same process as before, we derive the governing equations by filtering out every term in the original equation (1.1). Because of linearity of convolution, the only term that requires further comment is the nonlinear convection term. To this end, we define the residual-stress tensor  $\tau'_{ij}$  as the difference between the filtered product of velocities and the product of the filtered velocities, i.e.

$$\tau'_{ij} = \overline{U_i U_j} - \bar{U}_i \bar{U}_j. \quad (1.35)$$

We can then write the final form of the equations governing the filtered fields

$$\frac{D\bar{U}_i}{Dt} = \nu \frac{\partial^2 \bar{U}_i}{\partial x_j \partial x_j} - \frac{1}{\rho} \frac{\partial \bar{P}}{\partial x_i} - \frac{\partial \tau'_{ji}}{\partial x_j}, \quad (1.36)$$

$$\frac{\partial \bar{U}_i}{\partial x_i} = 0. \quad (1.37)$$

There is again an obvious similarity to the mean flow equations, with the residual-stress tensor playing the role of the Reynolds stresses. Although both sets of equations are structurally the same, we must be careful when drawing such analogies for two main reasons, stemming from the fact that the fields involved have different interpretations:

1. As already mentioned, the filtered field is still a random variable, and solving this set of equations can only be understood as computing one possible realization.
2. While both sets are incomplete, we need to look for different models for  $\tau_{ij}$  than we did for  $\langle u_i u_j \rangle$ , because they represent different concepts.

Of the models used to represent  $\tau_{ij}$ , the most notable is the Smagorinsky model [33], which takes the residual-stress tensor to be proportional to the filtered shear-stress tensor  $\bar{S}_{ij}$  through the eddy viscosity of residual motions  $\nu_r = \nu_r(t, \mathbf{x})$

$$\tau_{ij} = -2\nu_r \bar{S}_{ij}. \quad (1.38)$$

The residual eddy viscosity depends on the filter width  $\Delta$ , which is a quantity that parametrizes the filter  $G$ , characterizing the size of the filtered eddies.

Various filters can be used, but they all have to satisfy the normalization condition

$$\int_{\Omega} G(\mathbf{r}) \, d\mathbf{r} = 1. \quad (1.39)$$

The three simplest examples of one-dimensional filters are the box filter, sharp spectral filter, and the Gaussian filter, given by

$$G_b(r) = \frac{1}{\Delta} \chi_{(-\frac{1}{2}\Delta, \frac{1}{2}\Delta)}(r), \quad (1.40)$$

$$G_s(r) = \frac{\sin\left(\frac{r\pi}{\Delta}\right)}{r\pi}, \quad (1.41)$$

$$G_g(r) = \left(\frac{6}{\pi\Delta^2}\right)^{1/2} e^{-\frac{6r^2}{\Delta^2}}, \quad (1.42)$$

respectively. These filters represent the weight with which the filter averages the velocity field. While the box filter — averaging the field over a box — seems the simplest in its concept, for LES, it is important how the filter behaves in Fourier wavenumber space.

We define the transfer function  $\hat{G}(\kappa)$  corresponding to the filter as

$$\hat{G}(\kappa) = \int_{\Omega} G(r) e^{-i\kappa r} dr, \quad (1.43)$$

and the normalization condition translates to

$$\hat{G}(0) = \int_{\Omega} G(r) dr = 1. \quad (1.44)$$

For the aforementioned filters, the transfer functions are

$$\hat{G}_b(\kappa) = \frac{\sin\left(\frac{\kappa\Delta}{\pi}\right)}{\frac{1}{2}\kappa\Delta}, \quad (1.45)$$

$$\hat{G}_s(\kappa) = \chi_{\left(-\frac{\pi}{\Delta}, \frac{\pi}{\Delta}\right)}(\kappa), \quad (1.46)$$

$$\hat{G}_g(\kappa) = e^{-\frac{\kappa^2\Delta^2}{24}}. \quad (1.47)$$

The shapes of the filter and transfer functions can be seen in Figure 1.2. We can observe a certain symmetry between the box filter and sharp spectral filter, where the transfer function of one corresponds to the filter function of the other.

The meaning of the transfer functions becomes clear when we look at the Fourier representation of the filtered velocity. In the Fourier representation, the convolution becomes multiplication, so we get

$$\widehat{U}(\kappa) = \mathcal{F}\{\overline{U(x)}\} = \hat{G}(\kappa)\hat{U}(\kappa). \quad (1.48)$$

Here, we see that the significance of the transfer function is as a multiplication factor that diminishes certain wavenumbers. For example, the sharp spectral filter eliminates all wavenumbers larger than  $\frac{\pi}{\Delta}$  from the velocity field. The most important result of this is that the filtered field has significantly fewer wavenumbers that need to be modeled to fully resolve the flow, and we can, in turn, use a coarser grid ( $h = \Delta$  for the sharp spectral filter) without facing the downsides of unresolved turbulence.

LES in this form provides a significant improvement over DNS in terms of needed grid resolution and computational power. It also presents several challenges, such as the implementation of the filtering procedure and the choice of a suitable filter and residual stress model. To avoid these complications, simpler approaches have been developed.

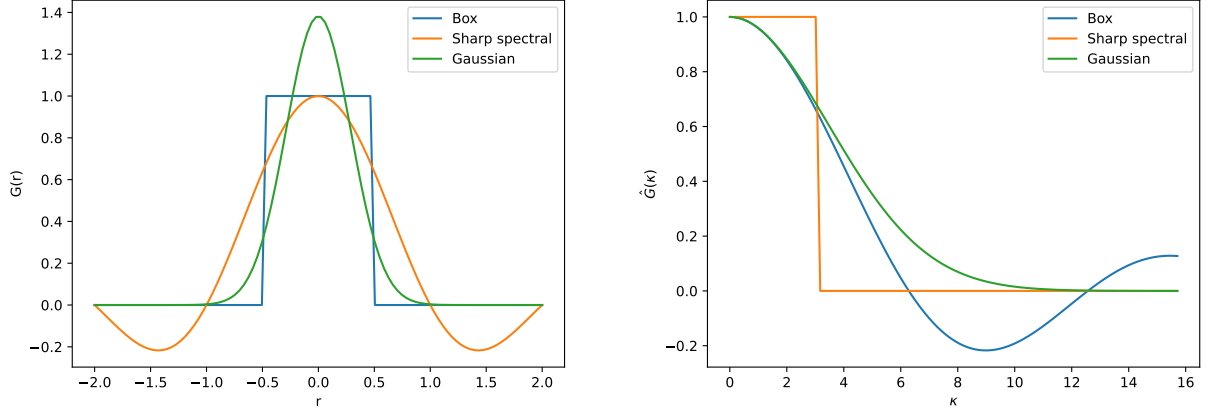


Figure 1.2: The filter functions and transfer functions for the box filter, sharp spectral filter, and Gaussian filter.

## 1.4 Spectral vanishing viscosity

One way to simplify LES is through a method called spectral vanishing viscosity (SVV). SVV is one of several methods that are sometimes referred to as implicit large eddy simulation (iLES). The central idea of these methods is similar to that of LES in so far as they also solve for filtered fields with higher wavenumbers diminished, but they do so without the need for explicit filtering or the choice of a specific model for the residual stress.

SVV was first introduced in [35], where it was used for the viscous regularization of the Burgers' equation in the context of spectral methods. We will follow the approach used in [18], with slight modifications to fit our equation.

We know that the smallest scale that needs to be resolved decreases with viscosity (i.e., the highest wavenumber increases as viscosity decreases). One way to decrease the number of wavenumbers we need to resolve would then be to increase the viscosity. However, this would affect the behavior of the small wavenumbers as well, which in turn changes important characteristics of our flow and might even lead to laminar flow. The main idea behind SVV is to increase the viscosity only for the high wavenumbers, leaving the small wavenumbers unaffected, thus preserving the large-scale behavior.

To achieve this, a new operator  $Q$  is introduced as an additional viscous term in the conservation of momentum equation (1.1)

$$\frac{DU_i}{Dt} = \nu \frac{\partial^2 U_i}{\partial x_j \partial x_j} + \nu_s \frac{\partial}{\partial x_j} \left[ Q \left( \frac{\partial U_i}{\partial x_j} \right) \right] - \frac{1}{\rho} \frac{\partial P}{\partial x_i}, \quad (1.49)$$

where  $\nu_s$  is the magnitude of the spectral vanishing viscosity. The operator  $Q$  should be chosen such that it is zero for the wavenumbers we wish to keep, and then increases with wavenumber to dampen the higher modes. One illustrative way to choose  $Q$  was shown in [18]. In that article,  $Q$  was chosen as a convolution operator with some viscosity kernel  $\hat{Q}$ . Now, assuming a periodic flow, we may discretize the equation using Fourier expansion. The convolution becomes multiplication, and the viscous terms take the form

$$\nu \frac{\partial^2 U_i}{\partial u_j \partial u_j} + \nu_s \frac{\partial}{\partial x_j} \left[ Q * \frac{\partial U_i}{\partial x_j} \right] = \nu \sum_{k=1}^{+\infty} k^2 (1 + \hat{Q}_k) \hat{u}_k(t) e^{ikx}. \quad (1.50)$$

Now it is easy to choose the transformed kernel  $\hat{Q}$  such that it satisfies our needs. For example, the choice made in the original article [35] was

$$\hat{Q} = \begin{cases} 0 & k \leq M \\ 1 & k > M \end{cases}. \quad (1.51)$$

**Remark.** *It is obvious how this model can be easily implemented for the spectral method using the Fourier basis. Different basis would also not pose a problem, as the operator  $Q$  can be defined only via its effect on the high-order terms, without using convolution to explicitly specify the effect of  $Q$  in the original equation.*

As already mentioned, the reason why SVV is considered iLES is that the viscous term dampens the high-order modes in the same way a LES filter would. One might then wonder what corresponds to the residual-stress tensor. The answer is clear from looking at equation (1.49). The stress tensor corresponds to the new viscous term, i.e.,

$$\tau_{ij}^r = \nu_s Q \left( \frac{\partial U_j}{\partial x_i} \right). \quad (1.52)$$

This means that both the filter and the residual-stress model are elegantly hidden in the choice of the spectral viscosity operator  $Q$ .

## 1.5 Passive scalar advection

Before moving on to the numerical aspects of this work, we will briefly discuss the dynamics of a passive scalar field in a turbulent flow. Assume a conserved scalar field  $C$ . This can represent either the concentration of some other fluid or a temperature variation. In our case, it will be the former, but in both cases, the value must be small enough not to affect the physical properties of the flow (hence the name "passive scalar"). The conservation law can be expressed in the form of the transport equation

$$\frac{DC}{Dt} = \Gamma \nabla^2 C, \quad (1.53)$$

where  $\Gamma$  is the diffusion coefficient. An important property of this equation is that the time evolution is bounded by the initial condition, i.e.,  $\min_{x \in \Omega} C_0(x) \leq \phi(t, x) \leq \max_{x \in \Omega} C_0(x)$  [31].

The scalar field is affected by the flow field via the material derivative. We want to know what effect turbulence has on the behavior of the scalar field. For this purpose, we might average the field in the same way we did with the velocity and pressure fields (i.e.,  $C = \langle C \rangle + c$ ), leading to the averaged equation

$$\frac{\langle D \rangle \langle C \rangle}{\langle D \rangle t} = -\nabla \cdot (\Gamma \nabla \langle C \rangle - \langle \mathbf{u}c \rangle). \quad (1.54)$$

As we can see, the equation is no longer closed, and a model for the turbulent scalar flux  $\langle \mathbf{u}c \rangle$  is needed. An example of such a model is the gradient-diffusion hypothesis [31], which sets  $\langle \mathbf{u}c \rangle = -\Gamma_t \nabla \langle C \rangle$ . This means that the scalar flux vector is aligned with the gradient of the scalar, in analogy to Fick's law of diffusion [8]. However, as it turns out, this model is too simple to correctly describe the turbulent transport of the scalar [36].

Although the transport equation is not directly affected by viscosity, the scalar field is influenced by the turbulent behavior. In the case of SVV, this means that the filtered velocity field will result in a different scalar field, which models the turbulent scalar flux in accordance with the filtered velocity field.



## Chapter 2

# Spectral/hp element method

This chapter will focus on the numerical methods used to solve the model described in the previous chapter. Specifically, it will cover a high-order version of the finite element method (FEM) [41], inspired by the spectral method (SM) [6], called the spectral/hp element method (SEM) [19]. This method is advantageous because it combines the fast convergence of spectral methods with the geometric versatility of the finite element method. It was originally introduced in [29], specifically for the purpose of simulating incompressible flow. Most of this chapter is adapted from [19].

### 2.1 Spectral method

We will start with the one-dimensional Burgers' equation with homogeneous Dirichlet boundary conditions, which can be considered a simplification of the Navier-Stokes equations

$$\frac{\partial U}{\partial t} + U \frac{\partial U}{\partial x} + \nu \frac{\partial^2 U}{\partial x^2} = 0, \quad (2.1)$$

$$U(\pm\pi) = 0. \quad (2.2)$$

The spectral method in its simplest form transforms the set of PDEs into an infinite set of ODEs via Fourier expansion and then approximates the solution by solving only for a finite number of modes,

$$U(t, x) = \sum_{k=1}^N \hat{U}_k(t) \cos(kx). \quad (2.3)$$

Substituting into the Burgers' equation, we get a set of ODEs

$$\frac{d\hat{U}_k}{dt} - k\hat{c}_k + \nu k^2 \hat{U}_k = 0, \quad (2.4)$$

where  $\hat{c}_k$  represents the Fourier expansion of the nonlinear term, which is usually modeled as a convolution of the two sums, i.e.

$$\hat{c}_k = \sum_{p=1}^N u_p u_{k-p}. \quad (2.5)$$

While this simple illustration suffices for our purposes, we should briefly mention the modern formulation of the spectral method, which is more sophisticated and general [6].

On a Hilbert space  $V$  with a finite subspace  $V_N$ , we look for an approximate solution  $U^N \in V_N$  such that

$$\left( \frac{\partial U^N}{\partial t} + U^N \frac{\partial U^N}{\partial x} + \nu \frac{\partial^2 U^N}{\partial x^2}, v^N \right) = 0 \quad \forall v^N \in V^N. \quad (2.6)$$

This form is called a Galerkin spectral method, and it is strikingly similar to the Galerkin formulation of FEM. One of the advantages of this formulation is that it allows us to use different bases for our expansion. Aside from the already mentioned trigonometric Fourier basis, various polynomial bases have been explored, including Chebyshev and Gauss-Legendre polynomials [6]. Polynomial bases will be further discussed in Section 2.3.

Among the biggest strengths of the spectral method is certainly its fast convergence rate. Unfortunately, it comes at the cost of the method not being geometrically flexible, as it is no simple task to find suitable bases of functions for more complicated geometries [19].

## 2.2 Finite element method

In contrast to the spectral method, the FEM is geometrically very flexible. The main idea of the method is to divide the geometry into  $N_{el}$  elements (of characteristic size  $h$ ) of simple geometry (e.g., triangles or tetrahedra) and define some polynomial functions on those elements, which are nonzero only on a few of the elements and zero everywhere else [41]. Unless stated otherwise, we will limit our description to the one-dimensional case for the sake of simplicity.

Limiting ourselves to a single element, upon which a finite subspace of polynomials is defined, we can write the discretized Burgers' equation as

$$\left( \frac{\partial U^N}{\partial t} + U^N \frac{\partial U^N}{\partial x} + \nu \frac{\partial^2 U^N}{\partial x^2}, v^N \right) = 0 \quad \forall v^N \in V^N. \quad (2.7)$$

We can see that this single-element discretization has the same equation as the spectral method. If the whole domain were made up of only one element, the methods would in fact be equivalent, assuming the same basis. Assuming there exists a suitable coordinate transform, all the elements can be represented by one standard element  $\Omega_{st}$ . The basis function  $\Phi_i(x)$  on the element  $\Omega_e$  can be represented by  $\phi_i(\xi)$  on  $\Omega_{st}$  as

$$\Phi_i(x) = \phi(\chi_e(x)), \quad (2.8)$$

where  $\chi_e : \Omega_e \rightarrow \Omega_{st}$  is the coordinate transform. The approximated solution on the whole domain can then be written in the form

$$U^\delta(t, x) = \sum_{i=0}^{N_{dofs}-1} \hat{U}_i(t) \Phi_i(x) = \sum_{e=1}^{N_{el}} \sum_{p=0}^P \hat{U}_p^e(t) \phi_p^e(\chi_e(x)). \quad (2.9)$$

Note that the number of summands is not necessarily the same in both sums. In the case of continuous Galerkin projection, we require  $U_\delta(t, x)$  to be continuous. This is ensured by having  $\hat{U}_0^e$  and  $\hat{U}_P^e$  represent the values on the boundary of the element, which restricts the coefficients with the continuity condition

$$\hat{U}_P^e = \hat{U}_0^{e+1} \quad \forall e \in \widehat{N_{el}-1}. \quad (2.10)$$

This condition ensures that in one dimension  $N_{dofs} = (P-1)N_{el} + 1$ . For higher dimensions, specific differences can vary due to each element having more neighbors.

FEM can be categorized into two types, depending on the type of convergence it employs. The  $h$ -type FEM achieves convergence by having a fixed set of polynomials for each element and making the elements smaller by division. In the limit  $h \rightarrow 0$ , the approximate solution converges to  $U$ . The  $p$ -type FEM, on the other hand, has a fixed grid and converges thanks to the increasing degree of the expansion  $P$  [19]. Combining these approaches, leads to  $hp$ -type methods, which use both mesh refinement and high order expansions.

## 2.3 Spectral/hp element method

The term "spectral element method" has come to describe a wide range of methods, but in the most general sense, it refers to a high-order FEM with a specific basis. While the standard FEM usually uses Lagrange polynomials through an equidistant set of points as the basis on each element, the spectral/hp element method offers more flexibility in the choice of basis functions. These can generally be categorized as either modal or nodal expansions. Modal expansions are those where the basis functions increase in order, while nodal expansions are those where all the basis functions are Lagrange polynomials of order  $P$ . The SEM nodal bases differ from the standard FEM bases by the specific choice of points through which the Lagrange polynomial is constructed. An example of a modal expansion is a moment expansion with the basis  $(x^p)_{p=0}^P$ .

As mentioned in the previous section, the continuity of the solution in nodal expansions is ensured by placing enough nodes on the boundaries of the element. With modal expansions, this is not as easily achieved, but it can be done, as shown in [19]. Furthermore, modal expansions can provide a different advantage. Consider the basis consisting of Legendre polynomials  $(L_p)_{p=0}^P$ , which can be found in [1].  $L_p$  have the advantage of satisfying the orthogonality relation

$$\int_{-1}^1 L_q(x)L_p(x) dx = \frac{2}{2p+1}\delta_{pq}. \quad (2.11)$$

Choosing this basis ensures that the mass matrix

$$\mathbb{M}_{ij} := \int_{\Omega} \Phi_i \Phi_j \quad (2.12)$$

is orthogonal.  $\mathbb{M}$  appears in the space-discretized ODEs in the term with the time derivative and therefore usually has to be inverted before attempting to solve the system. Having the mass matrix be diagonal simplifies the inversion and can reduce computation time. This comes at the cost of not having the continuity of the solution ensured.

Legendre polynomials are not the only ones to satisfy the orthogonality relation. Other Jacobi polynomials, if scaled correctly, will also produce these results, which is why they became popular in the context of SEM. We can use them to define a sophisticated modal basis

$$\phi_p(\xi) = \begin{cases} \frac{\xi-1}{2} & p = 0 \\ \frac{\xi^2-1}{4} P_{p-1}^{1,1}(\xi) & 0 < p < P, \\ \frac{\xi+1}{2} & p = P \end{cases} \quad (2.13)$$

where  $P_{p-1}^{1,1}$  is the Jacobi polynomial of order  $p-1$ . This expansion has several interesting properties. Specifically, the functions for  $0 < p < P$  are zero on the boundaries of the interval  $(-1, 1)$ , and we will call them interior modes. The rest are called boundary modes, which are one at one end of the interval and zero at the other. This division into interior and boundary modes is equivalent to that of standard FEM nodal bases and ensures that the continuity condition is just as easily satisfied.

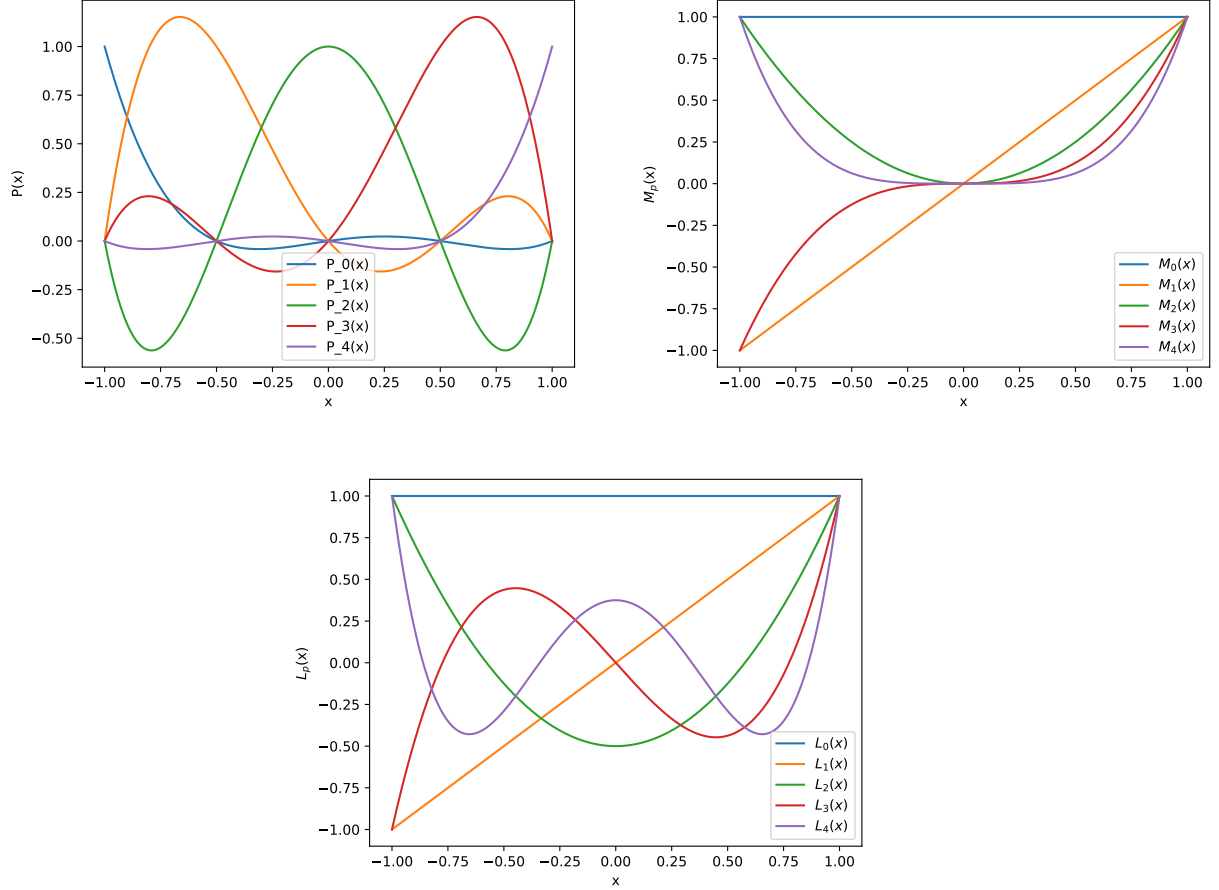
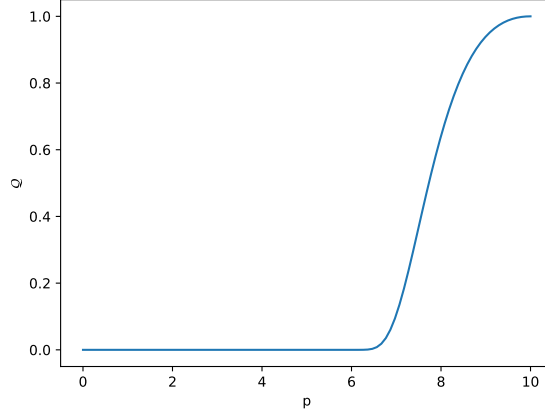


Figure 2.1: Shape of  $P = 5$  basis functions for equidistant nodal expansion (top left), moment expansion (top right), and the orthogonal Legendre expansion (bottom).

Another important quality of this modal expansion is that the Jacobi polynomials satisfy

$$\int_{-1}^1 (\xi^2 - 1) P_p^{1,1}(\xi) P_q^{1,1}(\xi) d\xi = C(p, q) \delta_{pq}. \quad (2.14)$$

This can be combined with the fact that Jacobi polynomials form a basis for a subspace of polynomials up to a given order. As a result, for  $0 < p < P$ , the basis function  $\phi_p$  is orthogonal to any polynomial of order less than  $p - 1$ . Specifically, they are orthogonal to any other basis function  $\phi_q$  with  $p - 2 > q$ . This means that the mass matrix  $\mathbb{M}$  will be tridiagonal, with the exception of the corners representing the modes  $p = 0$  and  $p = P$ . While this matrix structure is not as simple as if we were to use the fully orthogonal Legendre basis, it is simple enough to still significantly improve the inversion process, which, together with the simplification of the continuity condition, justifies the use of this basis.

Figure 2.2: The exponential SVV kernel  $Q$  for  $P = 10$ ,  $m = 5$ .

## 2.4 SVV for the spectral/hp element method

We will demonstrate the implementation of the SVV for the steady Navier-Stokes equation ( $\frac{\partial U}{\partial t} = 0$ ). We will use the above-defined modal basis for illustration, but the spectral vanishing viscosity (SVV) can be easily transformed into any other basis, just like any other linear operator. We follow the approach of [18], with slight modifications, such that our limit case is not inviscid, but rather has the desired viscosity  $\nu > 0$ . Consider the one-dimensional Burgers' equation with the added SVV term

$$\frac{\partial U}{\partial t} + U \frac{\partial U}{\partial x} = \nu \frac{\partial^2 U}{\partial x^2} + \nu_s \frac{\partial}{\partial x} \left[ Q \left( \frac{\partial U}{\partial x} \right) \right]. \quad (2.15)$$

We define the usual weak formulation notation

$$a(v, u) = \int_{\Omega} \frac{\partial v}{\partial x} \left[ \nu \frac{\partial u}{\partial x} + \nu_s Q \left( \frac{\partial u}{\partial x} \right) \right] dx, \quad (2.16)$$

$$f(v, u) = \int_{\Omega} v \left( \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} \right) dx. \quad (2.17)$$

Now, using the Galerkin approximation of FEM, we formulate the discretized equation, which must hold for all  $v^h \in V^h$ , where  $V^h$  consists of the span of all the basis functions  $\Phi_i^{(p)}$  (of which there are  $N_{dofs}$ ), with the superscript specifying the order of the polynomial mode

$$a(v^h, u^h) = f(v^h, u^h). \quad (2.18)$$

Expressing  $u^h$  as a linear combination of the basis functions:

$$u^h(t, x) = \sum_{i=1}^{N_{dofs}} \hat{u}_i(t) \Phi_i^{(p)}(x), \quad (2.19)$$

and taking the basis functions as the test functions, we get the set of equations

$$\sum_{j=1}^{N_{dofs}} \nu \hat{u}_j \left( \Phi_i^{(p)}, \Phi_j^{(q)} \right) + \sum_{j=1}^{N_{dofs}} \nu_s \hat{u}_j \left( \Phi_i^{(p)}, Q \left( \Phi_j^{(q)} \right) \right) + f \left( \Phi_i^{(p)}, u^h \right) = 0. \quad (2.20)$$



Figure 2.3: The effect of SVV in our simulations with parameters  $m = 0.75P$  (left) and  $m = 0.1P$  (right) on velocity magnitude at  $Re = 9000$ .

Now, we can see that for the implementation of SVV, it is sufficient to define the operator  $Q$  for the basis functions. The choice of  $Q$  is somewhat arbitrary, but we will use the one from [24], due to its simplicity. It is defined as

$$Q(\Phi_i^{(p)}) = Q(p)\Phi_i^{(p)}, \quad (2.21)$$

$$Q(p) = \begin{cases} 0 & \text{if } p \leq m, \\ e^{-\frac{(p-p)^2}{(p-m)^2}} & \text{if } p > m. \end{cases} \quad (2.22)$$

The SVV operator  $Q$  therefore increases the viscosity, which in turn dampens the high-order modes while being zero for the low-order modes, as shown in Figure 2.2. For illustration, the modal basis was discussed, but since any nodal basis function can be written as a linear combination of the modal basis functions, this method is applicable to any high-order FEM. The effect of SVV can be seen in Figure 2.3.

## 2.5 Gauss-Lobatto-Legendre basis

Besides the modal basis constructed in Section 2.3, a nodal basis frequently used in the spectral element method exists. This nodal basis consists of Lagrange polynomials through a set of node points, which are, unlike in the standard FEM case, not equidistant. Rather, the nodal points are chosen in such a way as to utilize the numerical integration scheme needed for evaluating the scalar products in the weak formulation [19]. First, we will investigate how polynomials can be numerically integrated.

Let us assume a polynomial  $h(x)$  of order  $P$  in  $(-1, 1)$ . The polynomial can be written as a linear combination of the equidistant nodal basis

$$h(x) = \sum_{p=0}^P h\left(-1 + \frac{2p}{P}\right) P_p(x). \quad (2.23)$$

The integral of  $h$  can then be written as a linear combination of the integrals of the basis polynomials

$$\int_{-1}^1 h(x) dx = \sum_{p=0}^P h\left(-1 + \frac{2p}{P}\right) \int_{-1}^1 P_p(x) dx = \sum_{p=0}^P h\left(-1 + \frac{2p}{P}\right) w_p. \quad (2.24)$$

This means that the integral of a polynomial can be evaluated exactly as a weighted sum of the function values at certain quadrature points. One of the most widely used quadrature schemes is the Gauss-Legendre quadrature, which has the quadrature points given by the zeros of the Legendre polynomials  $L_P$ . This quadrature formula is capable of exactly integrating any polynomial up to order  $2P + 1$ .

Unfortunately, the quadrature points are not well suited to construct the nodal basis, because none of them sit on the boundary of the interval. Therefore, we would lose the major advantage of the nodal basis, i.e., the possibility to be easily continuously connected to the neighboring elements. However, if we use only  $P - 2$  of the points for our basis and add the boundary points to have a complete set of  $P$  quadrature points, we get the Gauss-Legendre-Lobatto (GLL) quadrature. In doing so, we sacrifice some of the integration capability, as the GLL quadrature can exactly evaluate integrals of polynomials up to order  $2P - 1$ .

Now we construct a basis of Lagrange polynomials through the quadrature points of the GLL quadrature by satisfying the condition

$$\phi_i(\xi_j) = \delta_{ij}. \quad (2.25)$$

If we were now to compute the mass matrix using the GLL quadrature, we find that the product of two basis functions of order  $P$  is of order  $2P$  and therefore can only be approximated by the quadrature. However, the approximation is such that it makes the mass matrix orthogonal

$$\mathbb{M}_{ij} = (\phi_i, \phi_j) \approx \sum_{p=0}^P w_p \phi_i(\xi_p) \phi_j(\xi_p) = \sum_{p=0}^P w_p \delta_{ip} \delta_{jp} = w_i \delta_{ij}. \quad (2.26)$$

This proves to be a useful property when inverting the mass matrix, as mentioned earlier.

Both the nodal and modal bases, as defined in this chapter, are widely used in the spectral/hp element method. Their shapes can be seen in Figure 2.4. Applications usually require basis functions defined in more than one dimension. Unlike the standard FEM, which is usually formulated for triangular/tetrahedral elements, we will use the SEM on structured quadrilateral/hexahedral elements. The extension of the basis functions to more dimensions is done as a straightforward tensor product

$$\phi_{ij}(\xi_1, \xi_2) = \phi_i(\xi_1) \phi_j(\xi_2). \quad (2.27)$$

One only needs to be careful about the degree of the two-dimensional polynomials. Although the tensor product of polynomials of order  $P$  is of order  $2P$ , not all two-dimensional polynomials of order  $2P$  can be constructed in this way. The only two-dimensional polynomial spaces that can be fully constructed in this way are the polynomial spaces of order  $P$ .

## 2.6 Quasi 3D simulation

As was already stated, the main advantage of the SEM over spectral methods is its geometric flexibility. If the geometry of the problem in question has translational symmetry in one of the coordinates, the spectral method can be used in this coordinate to increase the rate of convergence [3]. For simplicity, we will only show the discretization for the linearized equation of conservation of momentum, i.e., the Stokes equation. The treatment of the nonlinear term would be the same as in Section 2.3.

Assume homogeneous Neumann boundary conditions in the spanwise direction. Further assume that the length of the domain in this direction is  $\pi$ . First, the equation is discretized in this direction (which we assume to be  $z$ )

$$U_i(t, x, y, z) = \sum_{k=0}^N \hat{U}_i^{(k)}(t, x, y) \cos(kz), \quad (2.28)$$

$$P(t, x, y, z) = \sum_{k=0}^N \hat{P}^{(k)}(t, x, y) \cos(kz). \quad (2.29)$$

**Remark.** We use only cosines in the expansion, in correspondence with the homogeneous Neumann boundary condition. Were we to use homogeneous Dirichlet boundary conditions, the equations would be written with sines. For periodic boundary conditions, as in the case of our simulations in Chapter 3, the expansion uses both sines and cosines.

Expressions (2.28),(2.29) are then substituted into the Stokes equation, along with the divergence-free condition

$$\sum_{k=0}^N \frac{\partial \hat{U}_1^{(k)}}{\partial t} \cos(kz) = \nu \sum_{k=0}^N (\nabla_{xy}^2 \hat{U}_1^{(k)} - k^2 \hat{U}_1^{(k)}) \cos(kz) - \frac{1}{\rho} \sum_{k=0}^N \frac{\partial \hat{P}^{(k)}}{\partial x} \cos(kz), \quad (2.30)$$

$$\sum_{k=0}^N \frac{\partial \hat{U}_2^{(k)}}{\partial t} \cos(kz) = \nu \sum_{k=0}^N (\nabla_{xy}^2 \hat{U}_2^{(k)} - k^2 \hat{U}_2^{(k)}) \cos(kz) - \frac{1}{\rho} \sum_{k=0}^N \frac{\partial \hat{P}^{(k)}}{\partial y} \cos(kz), \quad (2.31)$$

$$\sum_{k=0}^N \frac{\partial \hat{U}_3^{(k)}}{\partial t} \cos(kz) = \nu \sum_{k=0}^N (\nabla_{xy}^2 \hat{U}_3^{(k)} - k^2 \hat{U}_3^{(k)}) \cos(kz) + \frac{1}{\rho} \sum_{k=0}^N k \hat{P}^{(k)} \cos(kz), \quad (2.32)$$

$$\sum_{k=0}^N \left( \frac{\partial \hat{U}_1^{(k)}}{\partial x} + \frac{\partial \hat{U}_2^{(k)}}{\partial y} \right) \cos(kz) = \sum_{k=0}^N k \hat{U}_3^{(k)} \sin(kz). \quad (2.33)$$

Now, using the Galerkin projection (2.6) and the fact that the Fourier basis is orthonormal, we obtain four equations for each mode  $k = 0, \dots, N$

$$\frac{\partial \hat{U}_1^{(k)}}{\partial t} = \nu \left( \frac{\partial^2 \hat{U}_1^{(k)}}{\partial x^2} + \frac{\partial^2 \hat{U}_2^{(k)}}{\partial y^2} - k^2 \hat{U}_1^{(k)} \right) - \frac{\partial \hat{P}^{(k)}}{\partial x}, \quad (2.34)$$

$$\frac{\partial \hat{U}_2^{(k)}}{\partial t} = \nu \left( \frac{\partial^2 \hat{U}_1^{(k)}}{\partial x^2} + \frac{\partial^2 \hat{U}_2^{(k)}}{\partial y^2} - k^2 \hat{U}_2^{(k)} \right) - \frac{\partial \hat{P}^{(k)}}{\partial y}, \quad (2.35)$$

$$\frac{\partial \hat{U}_3^{(k)}}{\partial t} = \nu \left( \frac{\partial^2 \hat{U}_1^{(k)}}{\partial x^2} + \frac{\partial^2 \hat{U}_2^{(k)}}{\partial y^2} - k^2 \hat{U}_3^{(k)} \right) + k \hat{P}^{(k)}, \quad (2.36)$$

$$\frac{\partial \hat{U}_1^{(k)}}{\partial x} + \frac{\partial \hat{U}_2^{(k)}}{\partial y} = \frac{2k}{\pi} \sum_{m \neq k, m \pm k \text{ odd}} \frac{2m}{m^2 - k^2} \hat{U}_3^{(k)}. \quad (2.37)$$

These equations can then be further discretized in space using the SEM and integrated in time using a suitable scheme. As noted by [17], the divergence-free velocity equation couples the odd modes of the spanwise velocity to the even modes of the other elements of velocity and vice versa, which is why this method is sometimes referred to as a quasi-3D simulation.



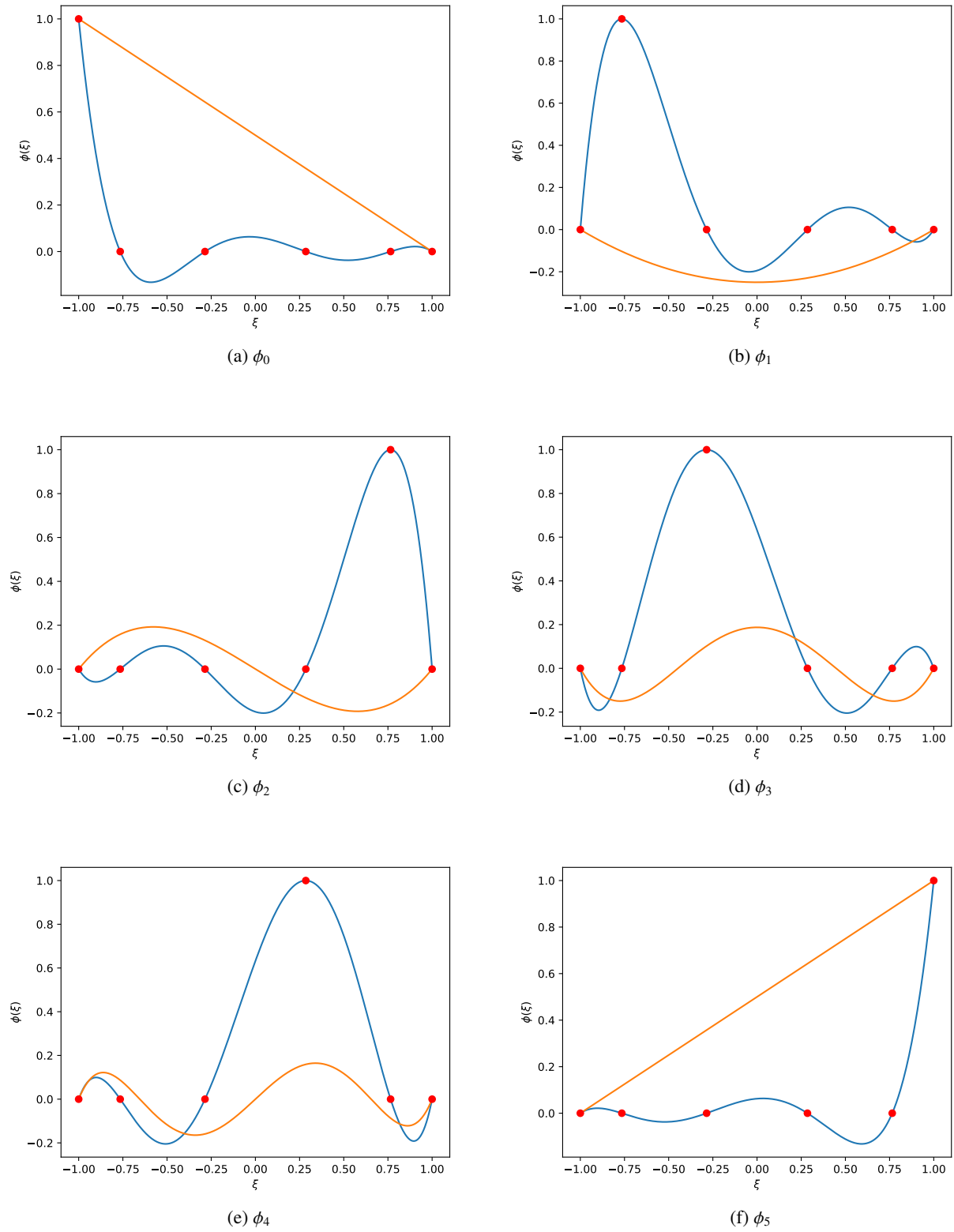


Figure 2.4: Modal (orange) and nodal (blue) basis functions  $\phi_0, \dots, \phi_5$  for  $P = 5$ , with the nodes highlighted in red.

## Chapter 3

# Numerical results

### 3.1 Problem set-up

As in any mathematical model, we need to find a balance between the accuracy of the model and its simplicity. Several geometries have been tested, but we ultimately settled on one that represents the important properties of the street network while remaining very simple. The final geometry for the purpose of simulation is a T-shaped domain with height and width of 2 in the  $xy$ -plane, extended homogeneously in the  $z$  direction, as shown in Figure 3.1. The width of the street canyon, the height of the building, and the width of the building are in the ratio 1:1:1, which is suitable for comparison with the experimental data [30][25].

The T-shape represents a street canyon with flat roofs. On the bottom of the street canyon and along the building surfaces, no-slip boundary conditions are chosen as a standard representation of walls and other solid surfaces

$$\mathbf{U}|_{\partial\Omega_1} = \mathbf{0}, \quad (3.1)$$

$$\frac{\partial P}{\partial \mathbf{n}}|_{\partial\Omega_1} = 0. \quad (3.2)$$

A rather difficult choice is the boundary condition on the upper side of the domain. Any choice of boundary condition here will be somewhat artificial compared to any physical street network or any wind tunnel measurement, because in those cases, there is a rather large open space above our limited domain. Simulating this open space, however, would come at the cost of a significant increase in required computing resources. Therefore, we chose a symmetric boundary condition, which represents how the flow would behave if the domain above the boundary were a symmetric reflection of the domain below the boundary. This means that the velocity component normal to the boundary must be zero, and the normal derivatives of all other fields must also be zero

$$\frac{\partial U_1}{\partial \mathbf{n}}|_{\partial\Omega_2} = 0, \quad (3.3)$$

$$U_2|_{\partial\Omega_2} = 0, \quad (3.4)$$

$$\frac{\partial U_3}{\partial \mathbf{n}}|_{\partial\Omega_2} = 0, \quad (3.5)$$

$$\frac{\partial P}{\partial \mathbf{n}}|_{\partial\Omega_2} = 0. \quad (3.6)$$

The advantage of this choice is that it ensures the flow on the boundary has a similar character to the flow in the rest of the domain, i.e., it is turbulent. The other often used boundary condition for such situation

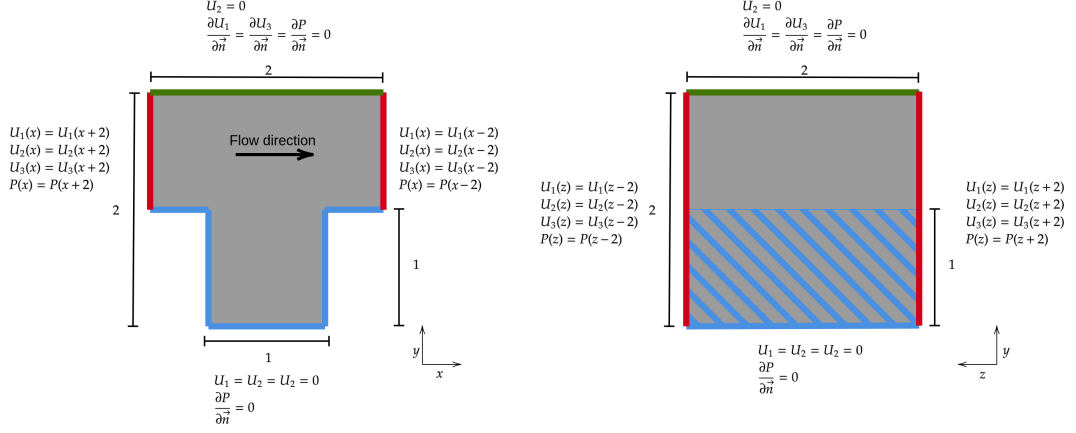


Figure 3.1: The simulation domain as viewed from the  $xy$ -plane and  $yz$ -plane, respectively. Blue lines represents the no-slip boundary condition, red lines the periodic boundary conditions and green line the symmetric boundary condition

would be constant flow imposed by Dirichlet condition, however in our geometry this leads to the flow becoming laminar.

The boundary condition at the left and right ends of the domain in the  $x$  direction is chosen to be periodic, i.e.

$$\mathbf{U}|_{\partial\Omega_3} = \mathbf{U}|_{\partial\Omega_4}, \quad (3.7)$$

$$P|_{\partial\Omega_3} = P|_{\partial\Omega_4}. \quad (3.8)$$

This ensures that the fully developed flow represents the flow across an infinite series of canyons, rather than a single canyon. Similarly, the boundary conditions at the ends of the domain in the  $z$  direction are chosen to be periodic, to represent an infinite street canyon

$$\mathbf{U}|_{\partial\Omega_5} = \mathbf{U}|_{\partial\Omega_6}, \quad (3.9)$$

$$P|_{\partial\Omega_5} = P|_{\partial\Omega_6}. \quad (3.10)$$

For the passive scalar, the choice of boundary conditions is straightforward, as they need to correspond to the boundary conditions of the other fields. That means periodic conditions in velocity and pressure require periodic conditions in the scalar field, no-slip boundary conditions require homogeneous Dirichlet conditions in the scalar field, and symmetric boundary conditions mean homogeneous Neumann conditions in the scalar field. In summary we have

$$\frac{\partial C}{\partial \mathbf{n}} \Big|_{\partial\Omega_1} = 0, \quad (3.11)$$

$$\frac{\partial C}{\partial \mathbf{n}} \Big|_{\partial\Omega_2} = 0, \quad (3.12)$$

$$C \Big|_{\partial\Omega_3} = C \Big|_{\partial\Omega_4}, \quad (3.13)$$

$$C \Big|_{\partial\Omega_5} = C \Big|_{\partial\Omega_6}. \quad (3.14)$$

The flow is started from a simple initial condition

$$U_1(0, \mathbf{x}) = 1 + Z(\mathbf{x}) \cdot 10^{-3}, \quad (3.15)$$

$$U_2(0, \mathbf{x}) = Z(\mathbf{x}) \cdot 10^{-3}, \quad (3.16)$$

$$U_3(0, \mathbf{x}) = Z(\mathbf{x}) \cdot 10^{-3}, \quad (3.17)$$

$$P(0, \mathbf{x}) = Z(\mathbf{x}) \cdot 10^{-3}, \quad (3.18)$$

$$C(0, \mathbf{x}) = 0. \quad (3.19)$$

where  $Z$  represents an added symmetric Gaussian white noise, which helps to develop turbulence. This flow is then accelerated by a constant volumetric force in the  $x$ -direction

$$\mathbf{F}(t, \mathbf{x}) = (0.0043, 0, 0), \quad (3.20)$$

until it reached a statistically stationary state. Only after sufficient time ( $T = 150$ , which corresponds to a particle moving with the characteristic passing 75 times through the domain) did we begin to collect the data. This means that the effect of the initial condition on the data should be negligible, and the main purpose of our choice was to approximate the stationary state so that the acceleration did not take too long.

## 3.2 Software tools used for the simulations

For the simulations, the spectral element method was used in the  $xy$ -plane, combined with the Fourier spectral method in the homogeneous  $z$ -direction. This coupling was done in the manner described in section 2.6. A 3D finite element mesh generator, Gmsh [10], was used to create the geometry and mesh, and the Spectral/hp element software, Nektar++ [5][27], was used to perform the simulations. The two-dimensional mesh was composed of structured quadrilateral elements (squares).

Aside from the quasi-3D simulations, other approaches, such as full 3D spectral element discretization, were explored. Such simulations require a structured hexahedral mesh, which can be generated using the *extrude* function of Gmsh. An example of a Gmsh script generating such geometry can be seen in the Appendix. However, this approach has proven to be inefficient for several reasons. The first reason is that the creation of such a structured mesh is rather complicated because it requires keeping track of all the new geometric elements created during the extrusion process, of which there are many for complicated geometries. The second reason is that the periodic conditions in Nektar++ require the mesh to have matching orientations on the periodic boundaries. This poses no problem in two dimensions or even in three dimensions with a single pair of periodic boundaries. However, it becomes a problem in three dimensions with two pairs of periodic boundaries, as in our case, because the corner hexahedrons, which are parts of both periodic boundaries, need to be oriented in such a way that they align with both periodic directions.

Neither of these reasons is impossible to overcome; however, they limit us to simple geometries. For such simple geometries, the quasi-3D method is more suitable due to its faster convergence properties. The script used to generate the mesh for the quasi-3D approach can be found in the Appendix.

Nektar++ employs a velocity correction scheme [11], which first solves the Poisson equation for pressure using an explicit scheme, and then solves the Helmholtz equation for each velocity component. These split equations are then handled by a second-order implicit-explicit (IMEX) solver [15]. GMRES is then used to solve the algebraic system of equations.

The SVV simulations were computed on the HELIOS cluster at the Faculty of Nuclear Sciences and Physical Engineering of the Czech Technical University in Prague. The simulations used 80 CPUs,

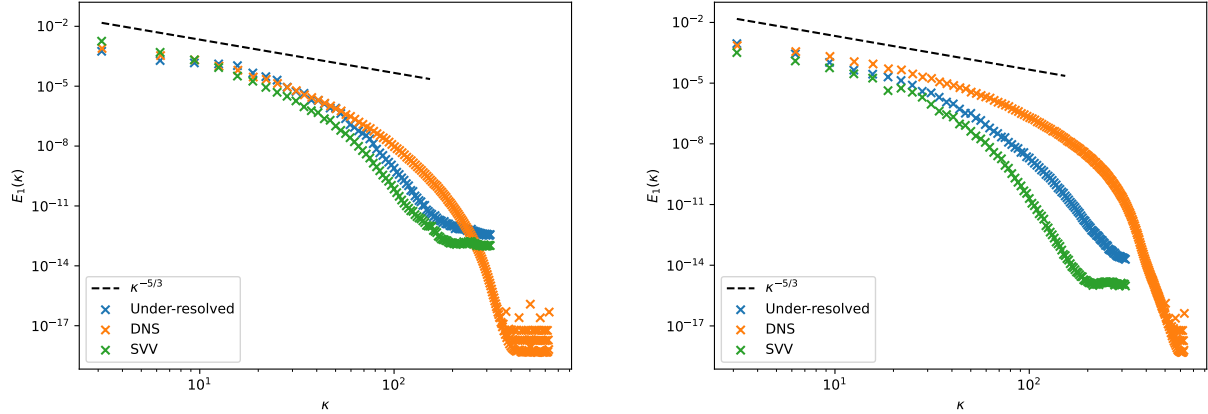


Figure 3.2: Comparison of the spectra in  $y = 1.2$  across different turbulence models for  $Re=4500$  (left) and  $Re=9000$  (right)

evenly distributed across four nodes, with each node equipped of 2x16-core AMD EPYC 7281@2.1GHz CPUs and 128 GB RAM. The reason for using 20 CPUs per node instead of all 32 is the way Nektar++ is parallelized, which requires the number of modes in the homogeneous direction to be an even multiple of the number of CPUs used (e.g., 80 CPUs for 160 homogeneous modes). The DNS and LES data were provided by my supervisor and were obtained using Extended Large-eddy Microscale Model [9].

An example of a Nektar++ configuration used for our simulations can be found in the Appendix

### 3.3 Comparison between iLES and DNS

The simulation was computed for Reynolds numbers  $Re=4500$  and  $Re=9000$ . To fully observe the effects of the SVV, we computed the flow both without SVV, which results in under-resolved turbulence, and with SVV, using a parameter of  $m = 0.5P$ . These results were compared to DNS using the finite difference method. For this purpose, a grid of size  $h = \frac{1}{30}$  was used, along with a nodal GLL polynomial expansion of order  $P = 4$  on each element in the  $xy$ -plane. In the span-wise ( $z$ ) direction, 160 Fourier modes were used.

The most interesting measurement is the one-dimensional spectrum, where the effect of SVV should be most prominent, for reasons discussed in Chapter 1. Indeed, in Figure 3.2, a sharp decrease can be observed in the SVV spectrum, especially for the higher Reynolds number. This is to be expected, since the flow with a lower Reynolds number has only a short inertial range, and the decrease due to viscosity starts at lower wavenumbers in the spectrum. This is not only the case for real viscosity but also for the spectral vanishing (and numerical) viscosity.

The difference in spectra is most easily visible in the turbulent kinetic energy. In Figure 3.3, we can see that the difference between the SVV and the under-resolved turbulence is more significant for the higher Reynolds number. In the upper part of the profile, representing the area above the canyon, where the turbulence can be considered statistically homogeneous, we observe that the turbulent kinetic energy is generally smaller in the SVV case than in the under-resolved case. This is in line with the energy spectra in Figure 3.2, where the energy at the wavenumbers is generally lower in the SVV case than in the under-resolved case. The same applies to the DNS, which has higher energy than the other two cases in both the spectrum and the  $y$ -profile.

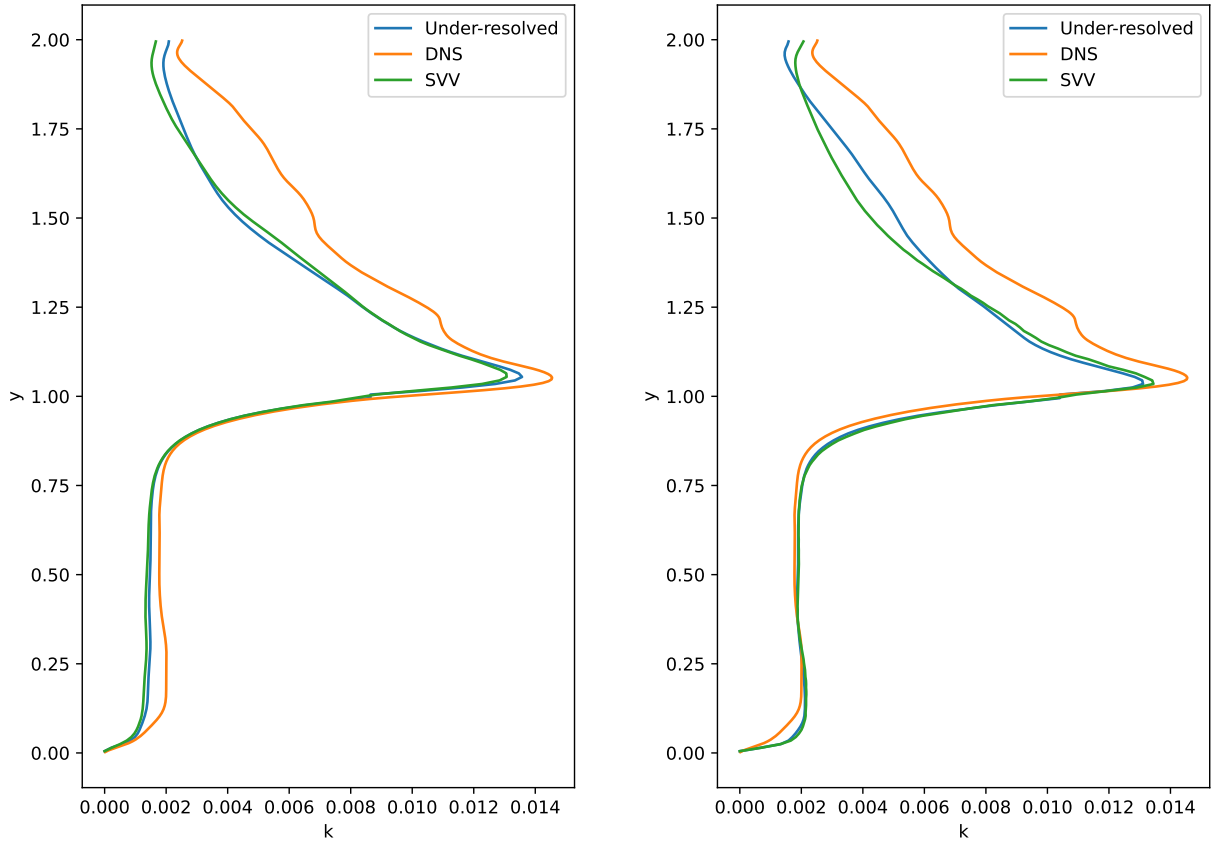


Figure 3.3: Comparison of  $y$ -profiles of the turbulent kinetic energy  $k$  for  $Re=4500$  (left) and  $Re=9000$  (right)

In the lower part of the graph, corresponding to the flow inside the street canyon, we see that the SVV and under-resolved simulations provide almost identical profiles, but neither is able to accurately simulate the energy of the vortex in the street canyon. This suggests that the flow inside the canyon contains a lot of energy in the smallest scales of turbulence. The exact effect is, however, hard to quantify since the flow is far from homogeneous, being strongly affected by the boundary conditions. Therefore, we cannot apply the same analysis as we did in the upper part of the domain.

It would be natural to extend this analysis further to higher Reynolds numbers. While the SVV case is stable even for high Reynolds numbers (up to  $Re=60000$  was successfully computed, but stability should be ensured even beyond that), the simulation becomes unstable in the under-resolved case for Reynolds numbers  $Re=10000$  and higher. A comparison with the under-resolved case is therefore not possible for higher Reynolds numbers. Another complication with high Reynolds numbers is that the inertial range of the spectrum is quite long, and the viscous dissipation starts only at very large wavenumbers (for very small eddies). This means that for DNS to fully resolve all the necessary scales, much higher spatial resolution is required.

	SVV (Re=9000 and Re=60000)	DNS (Re=9000)	LES (Re=60000)
Degrees of freedom	$6.9 \times 10^6$	$153.6 \times 10^6$	$1.2288 \times 10^9$
Number of CPUs	$4 \times 20$	$24 \times 16$	$24 \times 16$
Computation time	1,279 s	468 s	13,621 s
CPU time	102,320 s	179,712 s	5,230,464 s

Table 3.1: Performance of different simulation approaches. CPU time is the computation time multiplied by the number of CPUs. Computation time is defined as time needed to simulate one time unit.

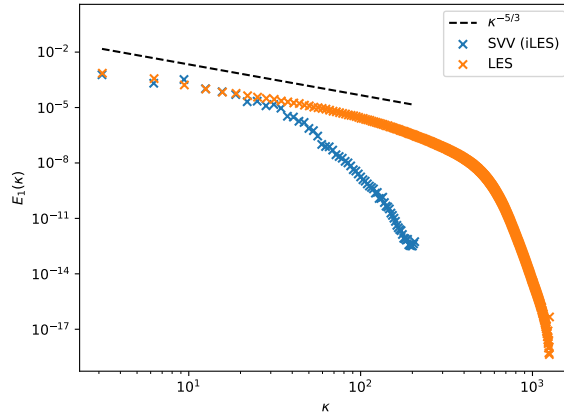


Figure 3.4: Comparison of spectra of SVV with  $m = 0.5P$  and LES for Re=60000

However, a comparison with traditional LES on a very fine grid, such that the filtering is very weak, can be made. Such LES will be very close to DNS in the wavenumbers we are concerned with and will only filter out the highest wavenumbers, which would need an unreasonably fine grid to be completely resolved. The comparison to such LES can be seen in Figure 3.4. Here, the inertial range is longer and clearly visible, as well as the effect of SVV, which shifts the dissipative range to lower wavenumbers.

The nature of the SVV allows for computing the flow on a significantly coarser grid, but the spectral element method is also more complicated for computation than the straightforward finite difference method. The coarser grid, with fewer degrees of freedom (DOFs), does not necessarily guarantee faster computation. The performance data are summarized in Table 3.1. The different simulation approaches are compared in terms of DOFs, computing time per time unit, and CPU time, which multiplies the number of CPUs by the computing time and should correspond to computing the simulation on a single CPU. Note that this last metric is only a rough estimate to compare the different approaches, as it does not take into account that parallel implementation differs from the serial implementation.

We can see that, for lower Reynolds numbers, the difference in CPU time is not significant, and the main advantage of the SVV approach is in the lower memory requirements due to the smaller number of DOFs. For higher Reynolds numbers, we see that the SVV approach, compared to LES (or DNS), also provides a significant advantage in CPU time. This means that the SVV method is stable even for coarser grid with fewer DOFs, but says nothing about the methods accuracy.

As we have seen, the SVV can be used to lower the range of wavenumbers that need to be resolved, allowing for a coarser grid. However, one needs to be careful when choosing the parameters of the SVV. If  $m$  is sufficiently low and  $\nu_s$  is sufficiently high, we found that oscillations in the spectrum may arise, i.e., when  $m \ll P$  and  $\nu_s \gg \nu$ .

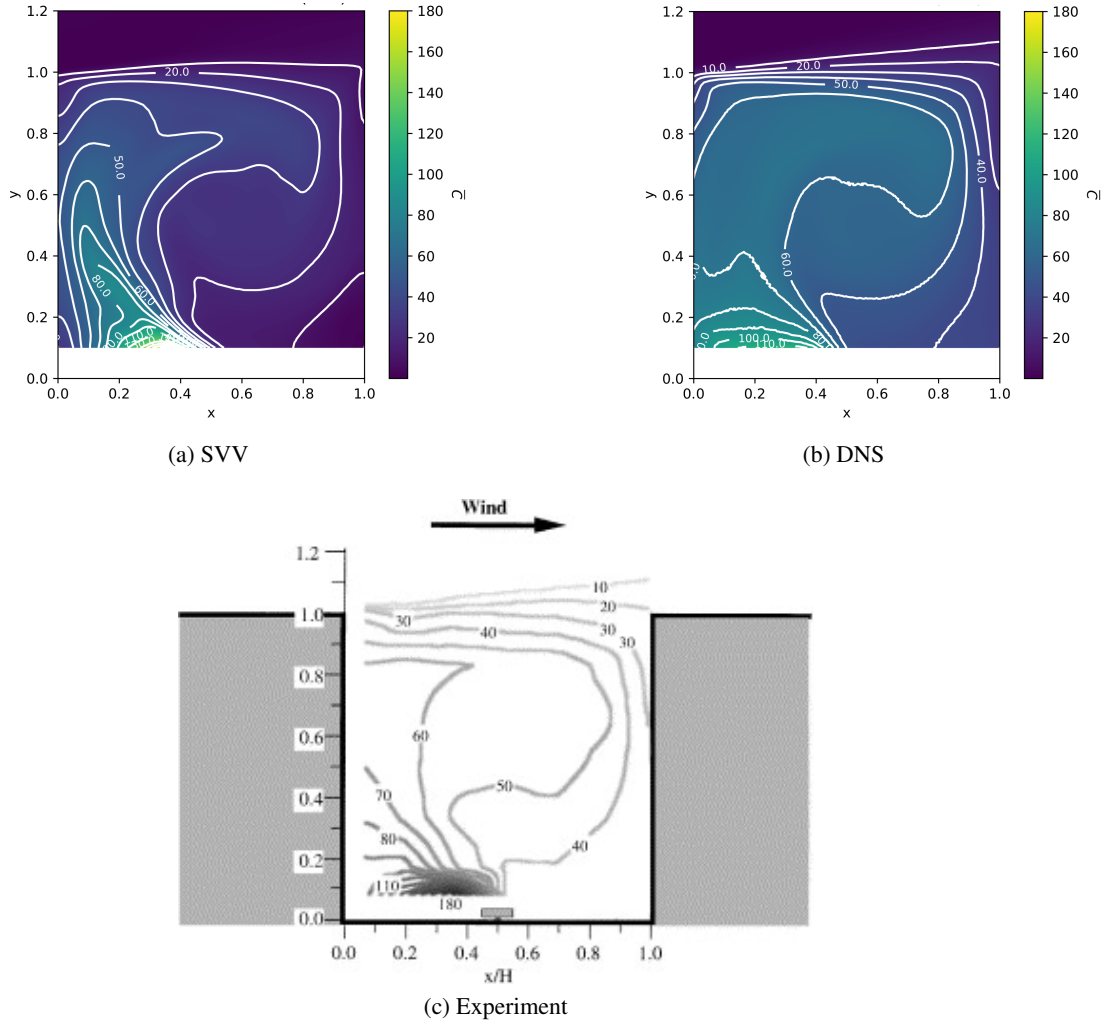


Figure 3.5: Comparison of time-averaged concentration of SVV simulated data (left), DNS data (right), and experimental data from [30] (bottom).

### 3.4 Passive scalar advection with line source

Besides DNS, we tested the method against experimental data. Specifically, we wanted to see how well it performs in passive scalar advection, which is affected by the turbulent flow. The simulated results were compared to the wind tunnel measurements from [30].

In the experiment, several street canyon buildings of height 60 mm were erected in sequence in the wind tunnel. On the floor of one of the canyons, a line source of gas was placed. After stabilizing into a stationary state, the concentration of this gas was measured at several points within the canyon.

For our simulation, we set  $Re = 4150$  to correspond with the experiment. In the region  $x < 0.2$ , the scalar passing through the periodic boundary condition was absorbed, as to correspond with the experiment, where no scalar was being carried from the previous street canyon. The concentration was then rescaled in the same manner as in [30] to account for different strengths of the source. The time-averaged concentration can then be seen in Figure 3.5.

As we can see, the shape of the isolines paints a similar structure of the concentration distribution.



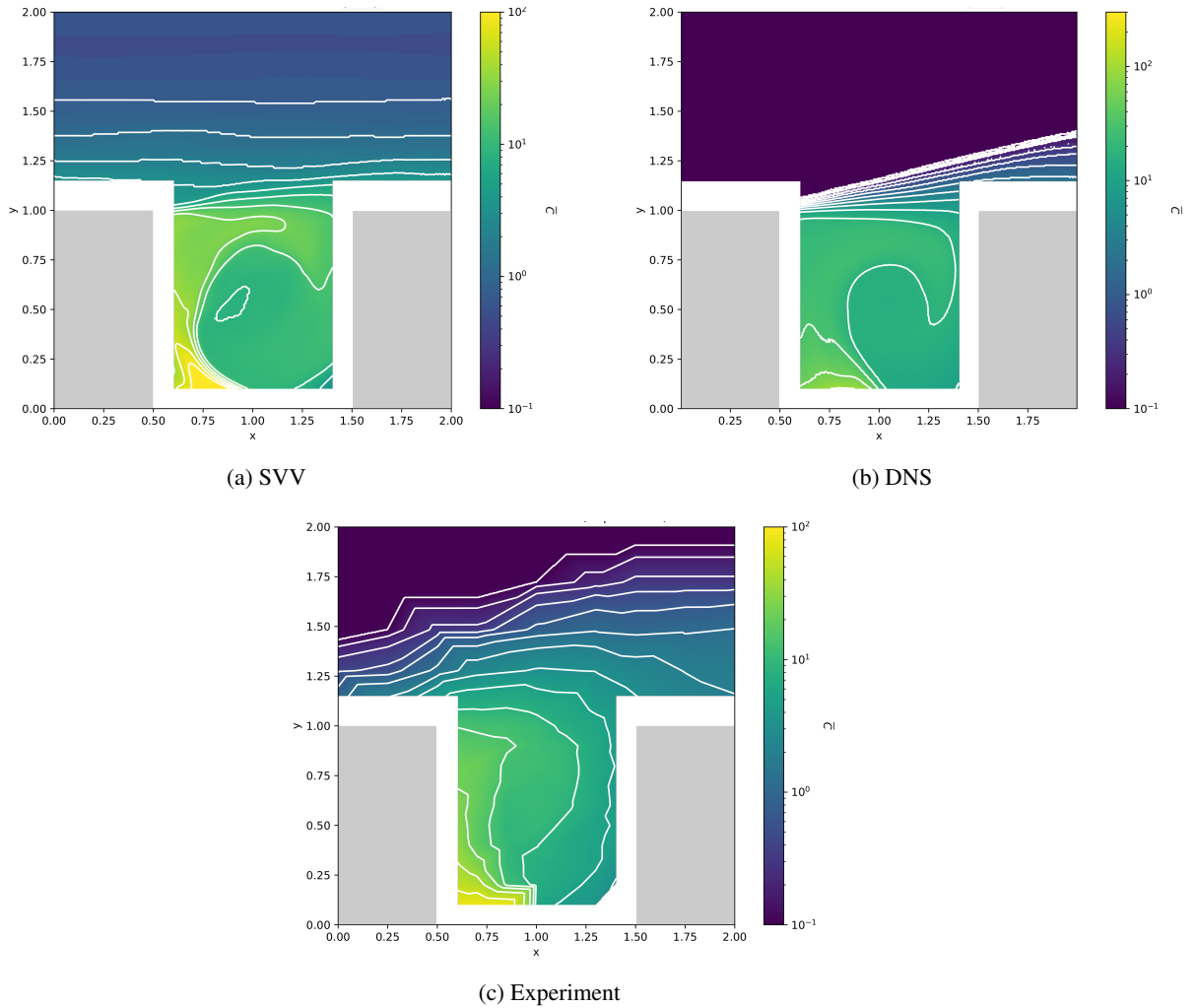


Figure 3.6: Concentration distribution in the  $xy$ -plane above the source for SVV (left), DNS (right) and the experimental data form [25] (bottom)

However, upon closer inspection, we can see that the levels do not match very well with the levels of the measured data. The DNS performs better in this respect because it produces a concentration more evenly spread in the canyon interior, which better matches the experimental distribution.

Unfortunately, due to the limited availability of experimental data, our analysis cannot go further than a comparison of data in the form of a single graph.

### 3.5 Passive scalar advection with point source

Another set of experimental data we tested our simulations against comes from [25]. These experiments differ from [30] mainly in that they used a single canyon with a point source of the passive scalar. For us, this mainly means that the flow above the canyon might differ from the experiment, due to the experimental setup not having other street canyons in front of the measured one. Other than that, the conditions are equivalent.

Due to the point source, we have to simulate a longer domain in the spanwise  $z$ -direction to avoid the effect of the scalar passing through the periodic boundaries in the spanwise direction. This spanwise contamination would make the simulation disanalogous to the experiment. The length of the canyon in the  $z$ -direction is therefore chosen as  $l = 4$  instead of  $l = 2$ , as pictured in Figure 3.1. The Reynolds number was chosen to correspond to the simulation, i.e.,  $Re = 4037$ .

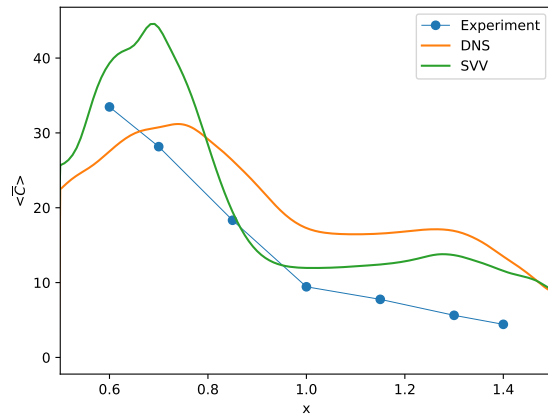
The time-averaged concentrations, which have been rescaled in the same manner as in [25], can be seen in Figure 3.6.

The most notable difference between the SVV, DNS, and the experimental data is the behavior above the canyon. While neither simulation manages to replicate the observed behavior in this area, each approach fails in a different manner. This can be attributed to two main causes. The first is the choice of boundary conditions, especially on the upper boundary, where the symmetric boundary condition is assumed. This boundary condition is, however, somewhat artificial and might not accurately represent the behavior in the experiment. The closer we get to this boundary, the more its effect will be visible. That is why this behavior is most prominent in the upper part of the domain, close to the boundary.

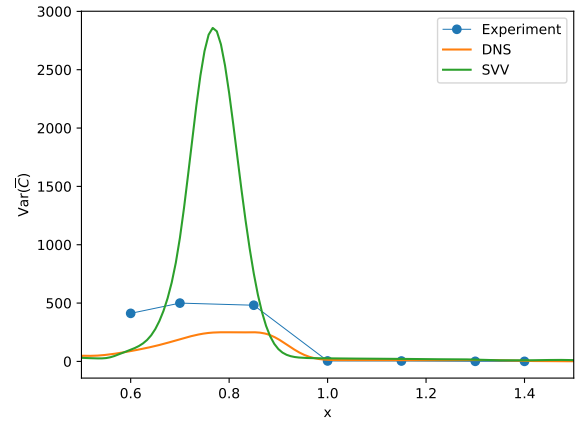
The second reason for the different behavior is only present for the SVV model, and it is the way in which the scalar is removed after passing through the periodic boundary. While in DNS it is directly erased, in the SVV case we use an absorbent layer, which gradually removes the scalar as it passes through the absorbent layer. While this process prevents a buildup of the scalar in the domain, it also leaves concentration in the left part of the domain in the time average. This residual concentration is amplified in the logarithmic scale of Figure 3.6.

For these reasons, we focus further analysis mostly on the area inside the street canyon. In Figure 3.7, we can see the mean and variance across  $y$  and  $x$  levels in the plane above the source, represented as  $x$  and  $y$  profiles, respectively. For the  $x$  profile, the statistical moments were taken from  $y \in [0, 1]$ , and for the  $y$  profiles, we took  $x \in [0.5, 1.5]$ . This choice ensures that the mean values inside the canyon are not affected by the values above the canyon and that the data is comparable to the data from [25].

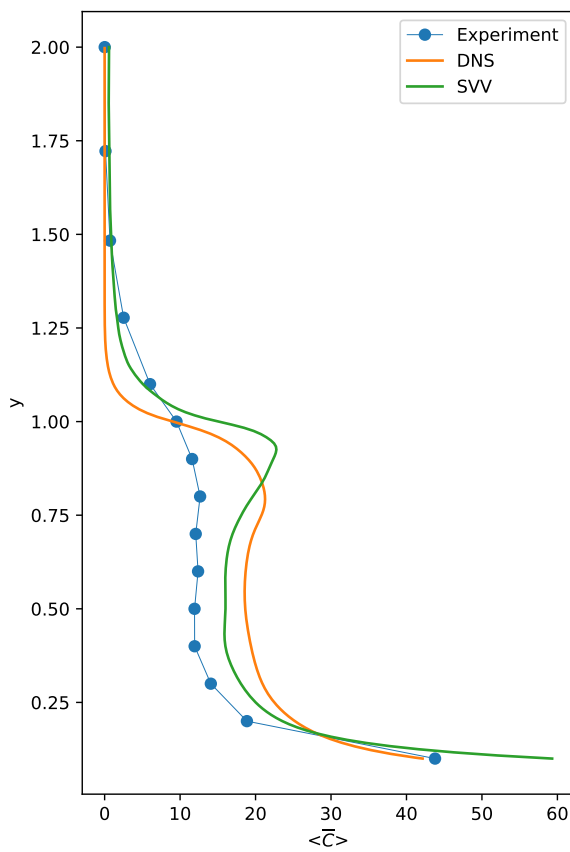
The mean profiles suggest that the SVV simulates the experimental data with reasonable accuracy, comparable to the DNS. The second moments, however, reveal that the DNS-simulated concentration is much more evenly distributed, especially in the  $y$  direction ( $x$  profile). This seems to better correspond to the experimental data, but the comparison here is limited because of the sparse nature of the experimental data. That is to say, the experimental data only provides about 10 data points in each line, which results in the statistical moments having lower statistical significance.



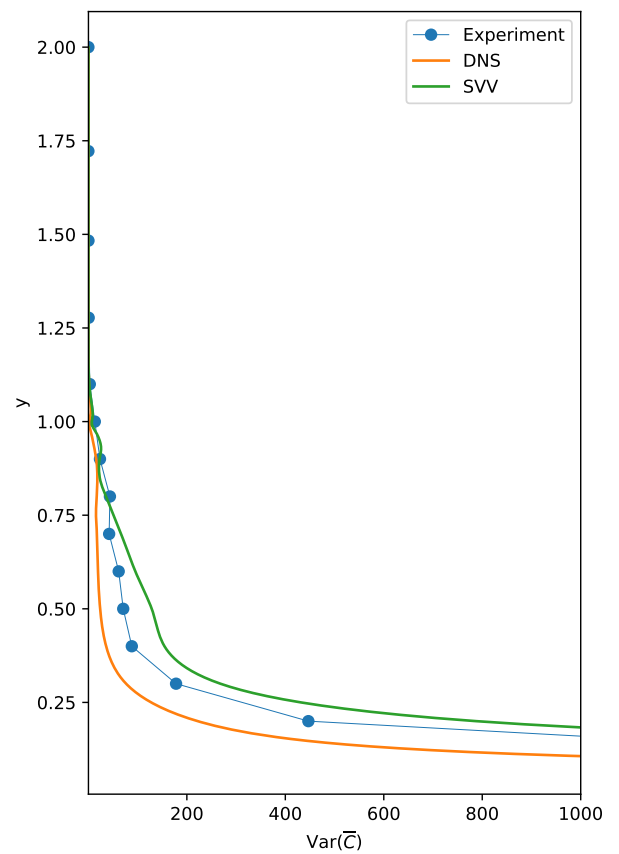
(a) Mean concentration  $x$ -profile



(b) Concentration variance  $x$ -profile



(c) Mean concentration  $y$ -profile



(d) Concentration variance  $y$ -profile

Figure 3.7: Profiles of first (left) and second central (right) statistical moments in  $x$  (top) and  $y$  (bottom) coordinates.

# Conclusion

We explored different approaches for modeling the turbulent transport of a passive scalar in a street network. We tested several geometries and boundary conditions until we settled on a suitable setup for the problem, which can be used for modeling turbulent transport in a street network. We then performed the simulations using the aforementioned methods.

Comparing the results of these simulations with other methods and experimental data, we found that spectral vanishing viscosity allows us to compute turbulent flow with high Reynolds numbers in a stable manner. Compared to direct numerical simulations using the finite difference method, it proved to be especially efficient for very high Reynolds numbers, where it was able to maintain the stability of the simulation even on a relatively coarse grid. The effect of spectral vanishing viscosity was observed on the one-dimensional energy spectrum in correspondence with theoretical predictions.

Two cases of pollutant transport were compared with experimental data: a line source and a point source. In both cases, our simulation provided a concentration distribution similar to that measured in the experiment. The concentrations computed using our method were not as accurate as those obtained by direct numerical simulation, but it was possible to compute them on significantly coarser grid, which made the simulations less computationally demanding.

Overall, the method described in this work has proven capable of efficiently simulating turbulent transport in a street canyon with reasonable accuracy on a coarse grid. Further research could focus on more complex geometries representing larger street networks or incorporating additional irregularities in the geometry of the buildings. Another remaining open question is the effect of different boundary conditions on the results of the simulation, particularly the boundary condition at the top of the domain, which currently has little physical justification.

# Appendix

## Nektar++ configuration

```
<NEKTAR>
<EXPANSIONS>
<E COMPOSITE="C[0]" NUMMODES="5" TYPE="GLL_LAGRANGE_SEM" FIELDS="u,v,w,p" />
</EXPANSIONS>
<FILTERS>
  <FILTER TYPE="ReynoldsStresses">
    <PARAM NAME="OutputFile">average</PARAM>
    <PARAM NAME="SampleFrequency">20</PARAM>
  </FILTER>

  <FILTER TYPE="Checkpoint">
    <PARAM NAME="OutputFile">output</PARAM>
    <PARAM NAME="OutputFrequency">7500</PARAM>
  </FILTER>

  <FILTER TYPE="HistoryPoints">
    <PARAM NAME="OutputFile">history</PARAM>
    <PARAM NAME="OutputFrequency">20</PARAM>
    <PARAM NAME="OutputOneFile">>true</PARAM>
    <PARAM NAME="box">10,5,10,0,2,1.2,1.3,0,2</PARAM>
  </FILTER>
</FILTERS>
<CONDITIONS>
  <SOLVERINFO>
    <I PROPERTY="SolverType" VALUE="VelocityCorrectionScheme"/>
    <I PROPERTY="EQTYPE" VALUE="UnsteadyNavierStokes"/>
    <I PROPERTY="Projection" VALUE="Galerkin"/>
    <I PROPERTY="TimeIntegrationMethod" VALUE="IMEXOrder2"/>
    <I PROPERTY="HOMOGENEOUS" VALUE="1D"/>
    <I PROPERTY="SpectralVanishingViscosity" VALUE="True" />
  </SOLVERINFO>

  <GLOBALSYSSOLNINFO>
    <V VAR = "u,v,w">
      <I PROPERTY="GlobalSysSoln" VALUE="IterativeStaticCond"/>
    </V>
  </GLOBALSYSSOLNINFO>
</CONDITIONS>
</NEKTAR>
```

```

    <I PROPERTY="Preconditioner" VALUE="Jacobi" />
    <I PROPERTY="IterativeSolverTolerance" VALUE="1e-8" />
    <I PROPERTY="NekLinSysMaxIterations" VALUE="10000" />
    <I PROPERTY="LinSysIterSolver" VALUE="GMRES" />
    <I PROPERTY="SuccessiveRHS" VALUE="8" />
  </V>
  <V VAR = "p">
    <I PROPERTY="GlobalSysSoln" VALUE="IterativeStaticCond" />
    <I PROPERTY="Preconditioner" VALUE="Jacobi" />
    <I PROPERTY="IterativeSolverTolerance" VALUE="1e-6" />
    <I PROPERTY="NekLinSysMaxIterations" VALUE="10000" />
    <I PROPERTY="LinSysIterSolver" VALUE="GMRES" />
    <I PROPERTY="SuccessiveRHS" VALUE="8" />
  </V>
</GLOBALSYSSOLNINFO>

<PARAMETERS>
<P> TimeStep      = 0.0005      </P>
<P> NumSteps      = 150000      </P>
<P> IO_InfoSteps  = 1000        </P>
<P> IO_CFLSteps   = 1000        </P>
<P> Re             = 4500        </P>
<P> Kinvis         = 1.0/Re      </P>
<P> HomModesZ     = 160         </P>
<P> LZ            = 2.0         </P>
<P> SVVCutoffRatio = 0.5 </P>
<P> SVVDiffCoeff  = 0.5 </P>
</PARAMETERS>

<VARIABLES>
<V ID="0">u</V>
<V ID="1">v</V>
<V ID="2">w</V>
<V ID="3">p</V>
</VARIABLES>

<BOUNDARYREGIONS>
<B ID="0">C[1]</B>
<B ID="1">C[2]</B>
<B ID="2">C[3]</B>
<B ID="3">C[4]</B>
</BOUNDARYREGIONS>

<BOUNDARYCONDITIONS>
  <REGION REF="0">
    <D VAR="u" VALUE="0.0" />
    <D VAR="v" VALUE="0.0" />

```

```

    <D VAR="w" VALUE=" 0.0 " />
    <N VAR="p" VALUE=" 0" USERDEFINEDTYPE="H" />
  </REGION>
  <REGION REF=" 1 ">
    <P VAR="u" VALUE=" [ 2 ] " />
    <P VAR="v" VALUE=" [ 2 ] " />
    <P VAR="w" VALUE=" [ 2 ] " />
    <P VAR="p" VALUE=" [ 2 ] " />
  </REGION>
  <REGION REF=" 2 ">
    <P VAR="u" VALUE=" [ 1 ] " />
    <P VAR="v" VALUE=" [ 1 ] " />
    <P VAR="w" VALUE=" [ 1 ] " />
    <P VAR="p" VALUE=" [ 1 ] " />
  </REGION>
  <REGION REF=" 3 ">
    <N VAR="u" VALUE=" 0" />
    <D VAR="v" VALUE=" 0" />
    <N VAR="w" VALUE=" 0" />
    <N VAR="p" VALUE=" 0" USERDEFINEDTYPE="H" />
  </REGION>
</BOUNDARYCONDITIONS>

<FUNCTION NAME=" BodyForce ">
  <E VAR="u" VALUE=" 0.0043 " />
  <E VAR="v" VALUE=" 0" />
  <E VAR="w" VALUE=" 0" />
</FUNCTION>

<FUNCTION NAME=" InitialConditions ">
  <F VAR="u" FILE=" ini . rst " />
  <F VAR="v" FILE=" ini . rst " />
  <F VAR="w" FILE=" ini . rst " />
  <F VAR="p" FILE=" ini . rst " />
</FUNCTION>
</CONDITIONS>
<FORCING>
  <FORCE TYPE=" Body ">
    <BODYFORCE>BodyForce</BODYFORCE>
  </FORCE>
</FORCING>
</NEKTAR>

```

## GMSH 3D mesh

lc = 1e-2;

```

n = 5;
l = 0.75;

// upper half base

p1 = newp; Point(p1) = {0, 0.25, 1, lc};
p2 = newp; Point(p2) = {0, 3.25, 1, lc};
p3 = newp; Point(p3) = {0, 3.25, 2, lc};
p4 = newp; Point(p4) = {0, 0.25, 2, lc};

p5 = newp; Point(p5) = {0, 0, 1, lc};
p6 = newp; Point(p6) = {0, 0, 2, lc};

p7 = newp; Point(p7) = {0, 3.5, 1, lc};
p8 = newp; Point(p8) = {0, 3.5, 2, lc};

l1 = newc; Line(l1) = {p1, p2};
l2 = newc; Line(l2) = {p2, p3};
l3 = newc; Line(l3) = {p3, p4};
l4 = newc; Line(l4) = {p4, p1};

l5 = newc; Line(l5) = {p1, p5};
l6 = newc; Line(l6) = {p5, p6};
l7 = newc; Line(l7) = {p6, p4};

l8 = newc; Line(l8) = {p3, p8};
l9 = newc; Line(l9) = {p8, p7};
l10 = newc; Line(l10) = {p7, p2};

Curve Loop(1) = {l1, l2, l3, l4};
Curve Loop(2) = {l5, l6, l7, l4};
Curve Loop(3) = {l8, l9, l10, l2};

Plane Surface(1) = {1};
Plane Surface(2) = {2};
Plane Surface(3) = {3};

Transfinite Curve{l1} = 12*n+2;
Transfinite Curve{l2} = 4*n+2;
Transfinite Curve{l3} = 12*n+2;
Transfinite Curve{l4} = 4*n+2;
Transfinite Curve{l5} = n+2;
Transfinite Curve{l6} = 4*n+2;
Transfinite Curve{l7} = n+2;
Transfinite Curve{l8} = n+2;
Transfinite Curve{l9} = 4*n+2;
Transfinite Curve{l10} = n+2;

```



```

Transfinite Surface {1} = {p1, p2, p3, p4};
Transfinite Surface {2} = {p1, p5, p6, p4};
Transfinite Surface {3} = {p3, p8, p7, p2};

Recombine Surface {1};
Recombine Surface {2};
Recombine Surface {3};

Extrude {1,0,0} {
    Surface {1}; Layers{ {3*n}, {1.0} };
    Recombine;
}

Extrude {1,0,0} {
    Surface {2}; Layers{ {3*n}, {1.0} };
    Recombine;
}

Extrude {1,0,0} {
    Surface {3}; Layers{ {3*n}, {1.0} };
    Recombine;
}

// bottom part base

p12 = newp; Point(p12) = {0, 0.25, 0.0, 1c};
p13 = newp; Point(p13) = {0, 0.0, 0.0, 1c};

p11 = newp; Point(p11) = {0, 3.25, 0.0, 1c};
p14 = newp; Point(p14) = {0, 3.5, 0.0, 1c};

l13 = newc; Line(l13) = {p12, p1};
l14 = newc; Line(l14) = {p5, p13};
l15 = newc; Line(l15) = {p13, p12};

l11 = newc; Line(l11) = {p2, p11};
l16 = newc; Line(l16) = {p11, p14};
l17 = newc; Line(l17) = {p14, p7};

Curve Loop(5) = {l14, l15, l13, l5};
Curve Loop(6) = {l16, l17, l10, l11};

Plane Surface(5) = {5};
Plane Surface(6) = {6};

```

```

Transfinite Curve{111} = 4*n+2;
Transfinite Curve{113} = 4*n+2;
Transfinite Curve{114} = 4*n+2;
Transfinite Curve{115} = n+2;
Transfinite Curve{116} = n+2;
Transfinite Curve{117} = 4*n+2;

Transfinite Surface{5} = {p1, p5, p12, p13};
Transfinite Surface{6} = {p2, p11, p14, p7};

Recombine Surface{5};
Recombine Surface{6};

Extrude {1,0,0} {
    Surface{5}; Layers{ {3*n}, {1.0} };
    Recombine;
}

Extrude {1,0,0} {
    Surface{6}; Layers{ {3*n}, {1.0} };
    Recombine;
}

120 = newc; Line(120) = {41, 40};
Curve Loop(7) = {120, 109, 12, 86};
Plane Surface(7) = {7};
Transfinite Curve{120} = 12*n+2;
Transfinite Surface{7} = {41, 40, 10, 9};
Recombine Surface{7};

Extrude {0.5,0,0} {
    Surface{7}; Layers{ {2*n}, {1.0} };
    Recombine;
}

Extrude {0.5,0,0} {
    Surface{32}; Layers{ {2*n}, {1.0} };
    Recombine;
}

Extrude {0.5,0,0} {
    Surface{76}; Layers{ {2*n}, {1.0} };
    Recombine;
}

```

```

Extrude {0.5,0,0} {
  Surface {54}; Layers{ {2*n}, {1.0} };
  Recombine;
}

```

```

Extrude {0.5,0,0} {
  Surface {104}; Layers{ {2*n}, {1.0} };
  Recombine;
}

```

```

Extrude {0.5,0,0} {
  Surface {126}; Layers{ {2*n}, {1.0} };
  Recombine;
}

```

```

Extrude {1,0,0} {
  Surface {171}; Layers{ {3*n}, {1.0} };
  Recombine;
}

```

```

Extrude {1,0,0} {
  Surface {215}; Layers{ {3*n}, {1.0} };
  Recombine;
}

```

```

Extrude {1,0,0} {
  Surface {193}; Layers{ {3*n}, {1.0} };
  Recombine;
}

```

```

Extrude {1,0,0} {
  Surface {237}; Layers{ {3*n}, {1.0} };
  Recombine;
}

```

```

Extrude {1,0,0} {
  Surface {259}; Layers{ {3*n}, {1.0} };
  Recombine;
}

```

```

Physical Surface (1) = {125, 136, 19, 99, 7, 149, 368, 268, 342, 113, 246, 356,
Physical Surface (2) = {281,325,369,347,303};
Physical Surface (3) = {1,2,3,5,6};
Physical Surface (4) = {45,206,294,91,224,334}; // front

```

```

Physical Surface (5) = {67,184,316,117,250,360}; // back
Physical Surface (6) = {49,210,298,27,166,276,63,180,312}; // top

Physical Volume (0) = {1, 2, 3, 4, 5, 6, 7, 8,
                      9, 10, 11, 12, 13, 14, 15, 16}; // volume

```

### GMSH quasi 3D mesh

```

lc = 1e-2;
n = 15;
h = 0.2;

p1 = newp; Point(p1) = {0, 1, 0, lc};
p2 = newp; Point(p2) = {2, 1, 0, lc};
p3 = newp; Point(p3) = {2, 2, 0, lc};
p4 = newp; Point(p4) = {0, 2, 0, lc};

p5 = newp; Point(p5) = {0.5, 0, 0, lc};
p6 = newp; Point(p6) = {1.5, 0, 0, lc};
p7 = newp; Point(p7) = {1.5, 1, 0, lc};
p8 = newp; Point(p8) = {0.5, 1, 0, lc};

l1 = newc; Line(l1) = {p1, p8};
l11 = newc; Line(l11) = {p8, p7};
l12 = newc; Line(l12) = {p7, p2};
l2 = newc; Line(l2) = {p2, p3};
l3 = newc; Line(l3) = {p3, p4};
l4 = newc; Line(l4) = {p4, p1};

Curve Loop(1) = {l1, l11, l12, l2, l3, l4};

Plane Surface(1) = {1};

Transfinite Curve{l1} = n+1;
Transfinite Curve{l11} = 2*n+1;
Transfinite Curve{l12} = n+1;
Transfinite Curve{l2} = 2*n+1;
Transfinite Curve{l3} = 4*n+1;
Transfinite Curve{l4} = 2*n+1;

Transfinite Surface{1} = {p1, p2, p3, p4};

Recombine Surface{1};

l5 = newc; Line(l5) = {p5, p6};
l6 = newc; Line(l6) = {p6, p7};

```

```

18 = newc; Line(18) = {p8, p5};

Curve Loop(2) = {15, 16, -111, 18};
Plane Surface(2) = {2};

Transfinite Curve{15} = 2*n+1;
Transfinite Curve{16} = 2*n+1;
Transfinite Curve{18} = 2*n+1;

Transfinite Surface{2} = {p5, p6, p7, p8};

Recombine Surface{2};

Physical Surface(0) = {1,2};
Physical Curve(1) = {11,112,15,16,18};
Physical Curve(2) = {12};
Physical Curve(3) = {14};
Physical Curve(4) = {13};

```

# Bibliography

- [1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York City, ninth dover printing, tenth GPO printing edition, 1964.
- [2] D.E. Aljure, O. Lehmkuhl, I. Rodríguez, and A. Oliva. Flow and turbulent structures around simplified car models. *Computers & Fluids*, 96:122–135, 2014.
- [3] C. H. Amon. Spectra element-fourier method for transitional flows in complex geometries. *AIAA Journal*, 31(1):42–48, 1993.
- [4] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge Mathematical Library. Cambridge University Press, 2000.
- [5] C.D. Cantwell, D. Moxey, A. Comerford, A. Bolis, G. Rocco, G. Mengaldo, D. De Grazia, S. Yakovlev, J.-E. Lombard, D. Ekelschot, B. Jordi, H. Xu, Y. Mohamied, C. Eskilsson, B. Nelson, P. Vos, C. Biotto, R.M. Kirby, and S.J. Sherwin. Nektar++: An open-source spectral/hp element framework. *Computer Physics Communications*, 192:205–219, 2015.
- [6] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods in Fluid Dynamics*. Computational Physics Series. Springer-Verlag, 1988.
- [7] P. Eichler, V. Fuka, and R. Fučík. Cumulant lattice boltzmann simulations of turbulent flow above rough surfaces. *Computers & Mathematics with Applications*, 92:37–47, 2021.
- [8] A. Fick. Ueber diffusion. *Annalen der Physik*, 170(1):59–86, 1855.
- [9] V. Fuka. Poissft – a free parallel fast poisson solver. *Applied Mathematics and Computation*, 267:356–364, 2015. The Fourth European Seminar on Computing (ESCO 2014).
- [10] C. Geuzaine and J.-F. Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.
- [11] J. L. Guermond and J. Shen. Velocity-correction projection methods for incompressible flows. *SIAM Journal on Numerical Analysis*, 41(1):112–134, 2003.
- [12] W. Hambli, J. Slaughter, F. F. Buscariolo, and S. Sherwin. Extension of spectral/hp element methods towards robust large-eddy simulation of industrial automotive geometries. *Fluids*, 7(3), 2022.
- [13] T. Heus, C. C. van Heerwaarden, H. J. J. Jonker, A. Pier Siebesma, S. Axelsen, K. van den Dries, O. Geoffroy, A. F. Moene, D. Pino, S. R. de Roode, and J. Vilà-Guerau de Arellano. Formulation of the dutch atmospheric large-eddy simulation (dales) and overview of its applications. *Geoscientific Model Development*, 3(2):415–444, 2010.

- [14] G. Hou, C. Chen, S. Qin, Y. Gao, and K. Wang. *Computational Fluid Dynamics: Finite Difference Method and Lattice Boltzmann Method*. Engineering Applications of Computational Methods. Springer Nature Singapore, Imprint: Springer, 2024.
- [15] T. J. R. Hughes and W. K. Liu. Implicit-explicit finite elements in transient analysis: Implementation and numerical examples. *Journal of Applied Mechanics*, 45(2):375–378, 06 1978.
- [16] H. J. Hussein, S. P. Capp, and W. K. George. Velocity measurements in a high-reynolds-number, momentum-conserving, axisymmetric, turbulent jet. *Journal of Fluid Mechanics*, 258:31–75, 1994.
- [17] M. Z. Islam and Y. Y. Tsui. Quasi-3d modeling and efficient simulation of laminar flows in microfluidic devices. *Sensors*, 16(10), 2016.
- [18] G. S. Karamanos and G. E. Karniadakis. A spectral vanishing viscosity method for large-eddy simulations. *Journal of Computational Physics*, 163(1):22–50, 2000.
- [19] G. Karniadakis and S. Sherwin. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford University Press, 06 2005.
- [20] A. N. Kolmogorov. Dissipation of energy in the locally isotropic turbulence. *C. R. Acad. Sci. URSS*, 32:16–18, 1941.
- [21] A. N. Kolmogorov. The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers. *C. R. Acad. Sci. URSS*, 30:301–305, 1941.
- [22] R. H. Kraichnan. The structure of isotropic turbulence at very high reynolds numbers. *Journal of Fluid Mechanics*, 5(4):497–543, 1959.
- [23] E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–148, March 1963.
- [24] Y. Maday, S. M. O. Kaber, and E. Tadmor. Legendre pseudospectral viscosity method for nonlinear conservation laws. *SIAM Journal on Numerical Analysis*, 30(2):321–342, 1993.
- [25] D. Marucci and M. Carpentieri. Stable and convective boundary-layer flows in an urban array. *Journal of Wind Engineering and Industrial Aerodynamics*, 200:104140, 2020.
- [26] F. Moukalled, L. Mangani, and M. Darwish. *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab*. Fluid Mechanics and Its Applications. Springer International Publishing, 2015.
- [27] D. Moxey, C. D. Cantwell, Y. Bao, A. Cassinelli, G. Castiglioni, S. Chun, E. Juda, E. Kazemi, K. Lackhove, J. Marcon, G. Mengaldo, D. Serson, M. Turner, H. Xu, J. Peiró, R. M. Kirby, and S. J. Sherwin. Nektar++: Enhancing the capability and application of high-fidelity spectral/hp element methods. *Computer Physics Communications*, 249:107110, 2020.
- [28] Š. Nosek, V. Fuka, L. Kukačka, Z. Kluková, and Z. Jaňour. Street-canyon pollution with respect to urban-array complexity: The role of lateral and mean pollution fluxes. *Building and Environment*, 138:221–234, 2018.
- [29] A. T. Patera. A spectral element method for fluid dynamics: Laminar flow in a channel expansion. *Journal of Computational Physics*, 54(3):468–488, 1984.

- [30] M. Pavageau and M. Schatzmann. Wind tunnel measurements of concentration fluctuations in an urban street canyon. *Atmospheric Environment*, 33(24):3961–3971, 1999.
- [31] S. B. Pope. *Turbulent Flows*. Cambridge University Press, 2000.
- [32] O. Reynolds. On the dynamical theory of incompressible viscous fluids and the determination of the criterion. *Philosophical Transactions of the Royal Society of London. (A.)*, 186:123–164, 1895.
- [33] J. Smagorinsky. General circulation experiments with primitive equations: I. the basic equations. *Monthly Weather Review*, 91(3):99–164, March 1963.
- [34] P. P. Sullivan, J. C. McWilliams, and Ch.-H. Moeng. A subgrid-scale model for large-eddy simulation of planetary boundary-layer flows. *Boundary-Layer Meteorology*, 71:247–276, 1994.
- [35] E. Tadmor. Convergence of spectral methods for nonlinear conservation laws. *SIAM Journal on Numerical Analysis*, 26(1):30–44, 1989.
- [36] S. Tavoularis and S. Corrsin. Experiments in nearly homogenous turbulent shear flow with a uniform mean temperature gradient. part 1. *Journal of Fluid Mechanics*, 104:311–347, 1981.
- [37] G. I. Taylor. The spectrum of turbulence. *Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences*, 164:476–490, 1938.
- [38] T. von Kármán. Progress in the statistical theory of turbulence. *Proceedings of the National Academy of Sciences*, 34(11):530–539, 1948.
- [39] I. Wygnanski and H. Fiedler. Some measurements in the self-preserving jet. *Journal of Fluid Mechanics*, 38(3):577–612, 1969.
- [40] W. Zhang, C. D. Markfort, and F. Porté-Agel. Wind-tunnel experiments of turbulent wind fields over a two-dimensional (2D) steep hill: Effects of the stable boundary layers. *Boundary-Layer Meteorology*, 188:441–461, 2023.
- [41] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals*. Butterworth-Heinemann, sixth edition, 2005.