Bounds on energy dissipation for stress driven shear flows

Andrew Olson Gallardo

CID 00826914

Department of Aeronautics
South Kensington Campus
Imperial College London
London SW7 2AZ
U.K.

Submitted in fulfillment of the requirements for the degree of
MEng Aeronautical Engineering of Imperial College London

May 30, 2017

This page was intentionally left blank.

Contents

List of Figures

Abstract

Based on the extended Doering-Constantin variational principle using the background field method, we present a correction factor that improves on previously existing analytical and computational bounds on the energy dissipation coefficient in turbulent stress driven shear flow by approximately a factor of five. Starting with a review of the variational method and the advances made by Wynn et al. by the use of semidefinite programming, we study the behaviour of the background field's spectral constraint and compare it with results from direct numerical simulations to obtain the correction factor. We also study the results from the simulations and the implementation by Wynn et al. to gain insight on the physical behaviour of stress driven shear flows. Finally, based on the lack of experimental data for this type of flows, we provide accurate data from direct numerical simulations in the hope that it will aid future research in this area.

I. Introduction

Turbulent flows present many interesting characteristics, both from a scientific and industrial point of view. The dramatic increase in diffusivity of momentum, mass and energy (heat) when a flow transitions from laminar to turbulent allows for more efficient heat exchangers, effective active cooling in turbine blades, and better sporting equipment such as lower drag golf balls, as a few examples [1].

Mathematically speaking, turbulence represents a very challenging problem. This is due to the equation that governs fluids: the Navier Stokes equation. When decomposed into a mean and fluctuating parts, and averaging (which turns the equation into the RANS form, or *Reynolds-averaged Navier Stokes*), highly nonlinear terms appear in the equation in the form of convective accelerations $-\rho\overline{u_i'u_j'}$, called Reynolds stresses $\boldsymbol{R}_{ij}$. These Reynolds stresses have an effect on the flow of a similar character as pressure of viscosity. However, they cannot be analytically solved: this is called *the problem of closure of turbulence*. A common approach is to use experimental data to model $\boldsymbol{R}_{ij}$ and therefore *close* the problem, allowing for a solution to be achieved. This method, although effective, relies on unproven assumptions about $\boldsymbol{R}_{ij}$ and experimental data.

Another alternative to try to solve this problem is to computationally solve Navier Stokes exactly. This is what *Direct Numerical Simulations* (DNS) do. As there is no modelling involved, this approach requires solving all length and time scales of turbulence exactly, which explains the computational cost of this method: the number of grid points in a 3D simulation scale with $Re^{9/4}$ [2]. The number of timesteps required also grows with a power law of the Reynolds number [3], which makes DNS unsuitable for most high-Re applications beyond $Re > 10^6$ [4].

What all these methods have in common is that they are trying to give us an insight into turbulence. Simulations rely on solving the equations of motion, but that is not the only option to extract information about turbulent flows. Since the 1970s, the idea of bounding key turbulent parameters (such as the dissipation coefficient C_ε) by using mathematically rigorous inequalities gained popularity thanks to the introduction of the Howard-Busse theory [5] and its further exploration and development.

This idea is promising, as these parameters can improve our understanding of certain turbulent flows. Providing an upper bound for C_ε as a function of a driving parameter (Reynolds number Re, Grasshoff number Gr) is providing an upper bound to the energy dissipation of the system and therefore an upper bound to the intensity of the turbulence as a function of the forcing parameter of the flow. It is because of this that these theories have gained traction.

Busse continued the work in this field, producing an asymptotic upper bound for the energy dissipation in a plane Couette flow in accordance with classical theories [4]. However, these still differ from experimental data by an order of magnitude. This solution by Busse was considered extremal, although there was no explicit proof for it. This result would not be improved on for 25 years, when Doering and Constantin [6], presented a different variational principle to obtain fundamental quantities characterising turbulent flows.

The need for a different variational principle was due to the classical theories that Busse relied on: they assume that the solution is unique and smooth, which is unproven. However, Doering et al. argue, it has not been ruled out that solutions of the incompressible Navier-Stokes equations can exhibit singularities starting from smooth initial conditions within a finite amount of time.

This new variational principle, however, does not require these assumptions, obtaining a more rigorous result. The initial applications of this principle could not compete with the results obtained by Busse, but when extended as seen in Ref. [7], this method obtains Busse's bound in the asymptotic regime.

The key to the method presented by Doering et al. is the use of what is called the *background method* [8]. This technique is based on the idea of decomposing the flow into a steady background flow ϕ and some disturbances $\tilde{\mathbf{u}}$, such that the background flow absorbs the inhomogeneous boundary conditions so that the disturbances only have to satisfy homogeneous boundary conditions. When applied to stress-driven shear flows, if the flow is laminar (classical laminar Couette flow, for example) ϕ is equal to the mean of the flow. However, in the turbulent regime this is not the case. Therefore, from now on, the term $\tilde{\mathbf{u}}$ will be referred to as *perturbations*, to distinguish it from the fluctuations term in the Reynolds Average Navier-Stokes equation $\mathbf{u}'$. The main difference between both terms is that the time average of the perturbations, $\overline{\tilde{\mathbf{u}}}$, is not necessarily zero.

To take advantage of this formulation, the bounds of the quantities to be obtained are expressed in terms of a functional $\mathcal{B}\{\phi\}$, with the condition of a quadratic function $\mathcal{Q}\{\phi\}$ being non-negative; condition known as spectral constraint. If a background field is chosen such as the positivity of the spectral constraint is guaranteed, then it is possible to obtain an upper bound for the chosen fundamental quantity (for example, the dissipation coefficient C_ε).

In other words, the key of the usefulness of the background method is that when the background component satisfies an energy stability-like equation as if it was the steady solution, then it provides upper bound for the momentum or heat transport for all solutions with the same boundary conditions.

The method employed in [7], an evolution from the method used by Doering and Constantin, consists of setting up a variational problem and solving the associated Euler-Lagrange system of equations to obtain the background flow. This method does present some drawbacks, as it requires careful computations for it to satisfy the spectral constraint [9]. This is further complicated in problems with imposed Neumann boundary conditions, as $\mathcal{B}\{\phi\}$ will depend on unknown values of ϕ in the boundary [10]. The background field method was first applied to the field of fluid mechanics on the traditional plane Couette flow for high Reynolds number [11]. However, the method has been used in the past by other fields such as quantum mechanics [12]. Doering et al argue in [7] that the method is applicable to all problems with a Neumann boundary condition (such as heat exchange), but in the recent past there have been challenges to whether the bounds obtained via this method are optimal for all problems. Nobili and Otto [13] argue that the bounds obtained with the background field applied to Rayleigh-Bernard convection are not optimal, with their main argument being the derivation of a tighter, yet mathematically rigorous bound using different methods.

Doering and Constantin obtained the background field through a variational principle, but in the recent past more innovative approaches have been successful at obtaining bounds for C_ε. One of the latest developments in the field is the use of semidefinite programming to solve an optimisation problem derived from the variational principle presented in [7], implemented by Wynn and Fantuzzi in [14]. By setting up the problem as an optimisation problem using semidefinite programming, the method avoids all complications arising from the *natural boundary conditions* in the Euler-Lagrange equations. This implementations seeks the minimisation of a functional subject to an especially formulated matrix containing the spectral constraint being semidefinite positive.

Semidefinite programming, a relatively novel technique, is proving very powerful in the field of convex optimisation [15]. A relatively recent development, it started gaining popularity in the 90s. Semidefinite programming can be seen as an abstraction of linear programs(LPs), representing a more general version of the problem. Instead of minimising or maximising an objective linear function subject to linear inequalities, a semidefinite programme (SDP) optimises a quadratic expression subject to a system of Linear Matrix Inequalities (LMIs) over the intersection of the cone of positive semidefinite matrices with an affine space.

Although semidefinite programs are much more general than linear programs, they are not significantly more difficult to solve. Most interior-point methods for linear programming have been generalized to semidefinite programs. As in linear programming, these methods have polynomial worst-case complexity and perform very well in practice [16]. Semidefinite programming has been applied to find approximate solutions to combinatorial optimization problems, such as the solution of the max cut problem with an approximation ratio of 0.87856 [17]. SDPs are also used in geometry to determine tensegrity graphs, and arise in control theory as LMIs.

Despite these attempts at obtaining optimal bounds, there is still a gap of approximately one order of magnitude between the experimental data and the bounds of C_ε obtained with the most recent methods. As the recently obtained bounds are considered to be at least near-optimal [14], there is no possible substantial improvement within the variational principle in [6]. Therefore, alternative methods must be sought, or adjustments must be made with empirical knowledge of the flow. This latter approach is difficult, due to the lack of experimental data for turbulent stress driven shear flow, as noted in [14].

Therefore, the aim of this dissertation is to investigate the mathematical behaviour of the bounds obtained by Wynn et al., as well as comparing the results to the results from Direct Numerical Simulations, in the search of possible improvements of the bound. Finally, we also provide accurate results from Direct Numerical Simulations for a range of Grasshoff numbers in the hope that it will help any future research of this topic.

The material is organised as follows. This introduction is followed by a Background and Methods section, where the mathematical framework is established. This includes the main equations for energy stability, as well as the formulation of the variational problem for C_ε. The software used for the optimisation problem and the Direct Numerical Simulations are also explored, with the most relevant equations being derived and any changes that have been made explained.

This section is followed by the Results and Discussion, where the behaviour of the spectral constraint is studied and scaling laws for C_ε are discussed. The main results from Direct Numerical Simulations are also discussed and compared to the plane Couette flow.

After a conclusion, two appendices are included. One with all the results from Direct Numerical Simulations for a range of Re_τ (the shear Reynolds number, $Re_\tau^2 = Gr$) between 60 and 150. The second appendix includes the modifications made to Diablo, the code used to perform Direct Numerical Simulations.

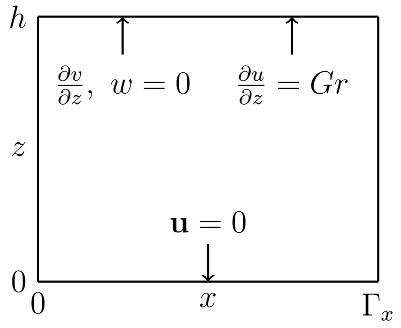
h
$\frac{\partial v}{\partial z}$, $w = 0$
$\frac{\partial u}{\partial z} = Gr$
z
$\mathbf{u} = 0$
0
0
x
Γ_x

Figure 1: Geometry for the problem of stress driven shear flow. The flow is periodic in both horizontal axes, and is driven by an applied shear stress Gr. The y direction is perpendicular to the paper, and periodic in Γ_y. The boundary conditions are also shown in the diagram.

II. Background and Methods

A. Mathematical framework

Energy stability in two dimensions

As an introduction to the problem, in this section the energy stability of the simplified two-dimensional stress driven shear flow problem is calculated. It will also serve to illustrate some of the points raised in the Introduction, as the method will follow the variational principle set by Doering et al. [7].

The energy evolution of a flow is of interest as it allows us to predict the critical point in which perturbations to the flow are no longer stable: in other words, the transition from a laminar to a turbulent state.

Firstly we introduce the problem. Consider a flow as seen in Figure (1), with a no-slip boundary condition in the bottom surface, periodic boundary conditions in the x direction and an applied shear stress on the top surface as a function of the non dimensional Grashoff number

$$Gr = \frac{\tau h^2}{\rho \nu^2}, \tag{1}$$

where τ is the shear stress applied at the top surface, h the vertical distance, ρ the density and ν the kinematic viscosity.

The analysis starts from the non-dimensional expression of Navier Stokes, after defining a velocity vector $\mathbf{u} = \boldsymbol{i}u + \boldsymbol{k}w$:

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p = \Delta\mathbf{u}, \\ \nabla\cdot\mathbf{u} = 0, \end{aligned} \tag{2}$$

with boundary conditions

$$\mathbf{u}|_{z=0} = 0, \quad w|_{z=1} = 0, \quad \left.\frac{dv}{dz}\right|_{z=1} = 0, \quad \left.\frac{\partial u}{\partial z}\right|_{z=1} = Gr, \tag{3}$$

obtained by considering the non dimensional variables

$$\boldsymbol{x} = \frac{\boldsymbol{x}_*}{h}, \quad t = \frac{t_*\nu}{h^2}, \quad \mathbf{u} = \frac{\mathbf{u}_* h}{\nu}, \quad p = \frac{p_* h^2}{\rho\nu^2}, \tag{4}$$

where the subscript $*$ denotes dimensional quantities. t represents time and p is pressure. This substitution is very useful as the forcing parameter of the equation, Gr, is present in the Neumann boundary condition, unlike other non-dimensionalisations that have a forcing parameter in the equation itself (usually Re).

The flow can be decomposed into a steady term $\mathbf{i}Grz$ depending on the forcing parameter and some fluctuations $\mathbf{u}'$. In the case of the simple steady, laminar flow, this decomposition leads to the Couette flow with $p = const$.

By substituting $\mathbf{u} = \mathbf{i}Grz + \mathbf{u}'$, denoting $\mathbf{u}' = \mathbf{i}u' + \mathbf{k}w'$ for the fluctuations, into 2, and noting that

$$\begin{aligned} \frac{\partial Grz}{\partial t} &= 0, \\ \mathbf{u}\cdot\nabla\mathbf{u} = \mathbf{i}\left(Grz(\frac{\partial u'}{\partial x} + w') + u'\frac{\partial u'}{\partial x} + w'\frac{\partial u'}{\partial z}\right)&, \\ +\mathbf{k}\left(Grz\frac{\partial w'}{\partial x} + u'\frac{\partial w'}{\partial x} + w'\frac{\partial w'}{\partial z}\right)&, \\ = Grz\frac{\partial \mathbf{u}'}{\partial x} + \mathbf{i}Grw'&, \\ \Delta\mathbf{u} = \mathbf{i}\frac{\partial^2 u'}{\partial x^2} + \mathbf{k}\frac{\partial^2 w'}{\partial z^2} = \Delta\mathbf{u}'&, \end{aligned}$$

the fluctuations' equations of motion are obtained, in the form of

$$\begin{aligned} \frac{\partial \mathbf{u}'}{\partial t} + \mathbf{u}'\cdot\nabla\mathbf{u}' + Grz\frac{\partial \mathbf{u}'}{\partial x} + \mathbf{i}Grw' + \nabla p = \Delta\mathbf{u}', \\ \nabla\cdot\mathbf{u}' = 0, \end{aligned} \tag{5}$$

with boundary conditions

$$\mathbf{u}'|_{z=0} = 0, \quad w'|_{z=1} = 0, \quad \left.\frac{\partial u'}{\partial z}\right|_{z=1} = 0. \tag{6}$$

Finally, the dot product of $\mathbf{u}'$ with 5 and an integration over the domain, and noting that

$$\int \mathbf{u}'\frac{\partial \mathbf{u}'}{\partial t}dxdz = \int \frac{1}{2}\frac{\partial \mathbf{u}'^2}{\partial t}dxdz = \frac{d}{dt}\frac{1}{2}\|\mathbf{u}'\|^2,$$
$$\mathbf{u}' \cdot \mathbf{i}Grw' = Gru'w',$$
$$\mathbf{u}' \cdot (\mathbf{u}' \cdot \nabla)\mathbf{u}' = |\nabla \mathbf{u}'|^2,$$

gives the energy evolution equation

$$\frac{d}{dt}\frac{1}{2}\|\mathbf{u}'\|^2 = -\int (|\nabla \mathbf{u}'|^2 + Gru'w')dxdz =: -\mathcal{E}\{\mathbf{u}'\}, \tag{7}$$

where $\frac{d}{dt}\frac{1}{2}\|\mathbf{u}'\|^2$ was obtained using the chain rule, and all the terms not mentioned vanish by repeated use of integration by parts. The term $\|\nabla \mathbf{u}\|^2$ is defined as

$$\|\nabla \mathbf{u}\|^2 = \sum_{i,j=1}^{3} \left\| \frac{\partial u_i}{\partial x_j} \right\|^2, \tag{8}$$

The quadratic expression $\mathcal{E}\{\mathbf{u}'\}$ gives us some insight on the energy stability of the solution: if $\mathcal{E}\{\mathbf{u}'\}$ is strictly positive for all divergence-free $\mathbf{u}'$ satisfying the boundary conditions in (6), then we can invoke Gronwall's lemma [18] . If ϕ satisfies

$$\phi'(t) \leq C(t)\phi(t), \tag{9}$$

where C is a nonnegative summable function, then Gronwall's lemma states

$$\phi(t) \leq \phi(0)exp\left(\int_0^t C(\tau)d\tau\right), \tag{10}$$

which implies exponential decay of $\|\mathbf{u}'\|^2$ with time. This therefore also implies the unconditional stability of the solution, as the energy of the fluctuations, $\|\mathbf{u}'\|^2$, is always bound by a finite value.

Using variational methods, Hagstrom and Doering, in [7], take (7) and obtain the eigenvalue problem

$$\lambda\Delta\Psi + \Delta^2\Psi + Gr\frac{\partial^2 \Psi}{\partial x \partial z} = 0, \tag{11}$$

with four boundary conditions

$$\left.\frac{\partial \Psi}{\partial x}\right|_{z=0,1} = 0, \quad \left.\frac{\partial \Psi}{\partial z}\right|_{z=0} = 0, \quad \left.\frac{\partial^2 \Psi}{\partial z^2}\right|_{z=1} = 0, \tag{12}$$

where Ψ is the *streamfunction* satisfying $\partial\Psi/\partial x = w'$ and $\partial\Psi/\partial z = -u'$. Due to to the system's periodicity in x, we can write the Fourier transformed variable $\Psi = \sum_k \hat{\Psi}_k e^{ikx}$ where $k = 2\pi n/\Gamma_x$ for integer $n = (-\infty, \infty)$. Substituting, the problem becomes the fourth-order ordinary differential equation

$$\lambda\left(\frac{\partial^2\hat{\Psi}}{\partial z} - k^2\hat{\Psi}\right) = -\left(\frac{\partial^4\hat{\Psi}}{\partial z^4} - 2k^2\frac{\partial^2\hat{\Psi}}{\partial z^2} + k^4\hat{\Psi}\right) - iGrk\frac{\partial\hat{\Psi}}{\partial z}, \tag{13}$$

with

$$\hat{\Psi}|_{z=0,1} = 0, \quad \left.\frac{\partial\hat{\Psi}}{\partial z}\right|_{z=0} = 0, \quad \left.\frac{\partial^2\hat{\Psi}}{\partial z^2}\right|_{z=1} = 0. \tag{14}$$

In the context of this variational principle and energy stability, the eigenvalues will all be positive if the flow is energy stable. By finding the value of Gr for which all eigenvalues λ are positive, we obtain the critical Gr of the flow. Beyond this critical Gr, the solution becomes unstable. In other words, the flow becomes turbulent. The critical Grashoff number is calculated to be 139.5 for two dimensions and 51.73 in three dimensions.

This problem illustrates the potential of the background field within the context of a variational principle to obtain bounds for key turbulent parameters.

Dissipation coefficient in three dimensions

Now we derive a bound for the dissipation coefficient C_ε, also known as the friction coefficient. Using the background method, and following the mathematical principles outlined in [7], we derive a functional $\mathcal{B}\{\phi\}$ as a function of the background field that represents a lower bound for the top layer velocity, which in turn provides an upper bound for the dissipation coefficient.

Dissipation scaling is traditionally accepted to be $\varepsilon \sim U^3/L$, where ε is the kinetic energy dissipation rate, U is the velocity and L is the integral length scale. J.C. Vassilicos [19] makes a comprehensive review on the validity of this assumption applied to boundary free turbulent flows, governed by Richardson-Kolmogorov cascades. We in turn investigate bounds for the dissipation specifically for stress driven shear flows, given the lack of available literature for this type of flows.

In a similar fashion as for the energy stability, we start from the non dimensional Navier-Stokes equation ((2)), but this time we emply the background method thanks to the substitution is $\mathbf{u} = \mathbf{i}\phi(z) + \tilde{\mathbf{u}}$, where $\tilde{\mathbf{u}} = \mathbf{i}\tilde{u} + \mathbf{j}\tilde{v} + \mathbf{k}\tilde{w}$. Noting that

$$\begin{aligned}\frac{\partial \mathbf{u}}{\partial t} &= \frac{\partial \tilde{\mathbf{u}}}{\partial t},\\ \mathbf{u}\cdot\nabla\mathbf{u} &= \tilde{\mathbf{u}}\cdot\nabla\tilde{\mathbf{u}} + \phi\frac{\partial \tilde{\mathbf{u}}}{\partial x} + \tilde{w}\frac{\partial \phi}{\partial z},\\ \Delta\mathbf{u} &= \Delta\tilde{\mathbf{u}} + \mathbf{i}\frac{\partial^2\phi}{\partial z^2},\end{aligned}$$

we obtain

$$\begin{gathered}\frac{\partial \tilde{\mathbf{u}}}{\partial t} + \tilde{\mathbf{u}}\cdot\nabla\tilde{\mathbf{u}} + \phi\frac{\partial \tilde{\mathbf{u}}}{\partial x} + \tilde{w}\frac{\partial \phi}{\partial z} + \nabla p = \Delta\tilde{\mathbf{u}} + \mathbf{i}\frac{\partial^2\phi}{\partial z^2}\\ \nabla\cdot\tilde{\mathbf{u}} = 0.\end{gathered} \tag{15}$$

with homogeneous boundary conditions

$$\tilde{\mathbf{u}}|_{z=0} = 0, \quad \tilde{v},\tilde{w}|_{z=1} = 0, \quad \left.\frac{\partial \tilde{u}}{\partial z}\right|_{z=1} = 0 \tag{16}$$

The substitution $\mathbf{u} = \mathbf{i}\phi(z) + \tilde{\mathbf{u}}$ decomposes the velocity field into a background flow $\phi(z)$ and some fluctuations $\tilde{\mathbf{u}}$. The background velocity is set so that it satisfies $\phi|_{z=0} = 0$ and the inhomogeneous boundary condition $d\phi/dz|_{z=1} = Gr$, allowing (15) to have homogeneous boundary conditions.

To progress towards obtaining $\mathcal{Q}$, we perform the space time average of (15) and obtain

$$\begin{aligned}0 = -\left\langle\frac{d|\tilde{\mathbf{u}}|^2}{dt}\right\rangle - \left\langle\frac{d\phi}{dz}\tilde{u}\tilde{w}\right\rangle + \left\langle\tilde{u}\frac{d^2\phi}{dz^2}\right\rangle\\ = \left\langle\frac{d|\tilde{\mathbf{u}}|^2}{dt}\right\rangle - \left\langle\frac{d\phi}{dz}\tilde{u}\tilde{w}\right\rangle +\\ + Gr(\tilde{u}(1) - \phi(1)) - \left\langle\frac{\partial\tilde{u}}{\partial z}\frac{d\phi}{dz}\right\rangle,\end{aligned} \tag{17}$$

by invoking the chain rule on $\langle\tilde{u}\frac{d^2\phi}{dz^2}\rangle$, knowing that $\overline{\mathbf{u}} = \overline{\tilde{\mathbf{u}}} + \mathbf{i}\phi$ and that $\|\tilde{\mathbf{u}}\|$ is bounded and therefore $\langle d|\tilde{\mathbf{u}}|^2/dt\rangle = 0$. The boundness of $\|\tilde{\mathbf{u}}\|$ is proven in [7].

Substituting $\overline{\mathbf{u}}(z) = \overline{\tilde{\mathbf{u}}}(z) + \mathbf{i}\phi(z)$ into $\langle|\nabla\mathbf{u}|^2\rangle$ and expanding gives

$$\langle|\nabla\mathbf{u}|^2\rangle = \langle|\nabla\tilde{\mathbf{u}}|^2\rangle + \left\langle\left(\frac{d\phi}{dz}\right)^2\right\rangle + \left\langle 2\frac{\partial\tilde{u}}{\partial z}\frac{d\phi}{dz}\right\rangle. \tag{18}$$

Finally, taking a linear combination with (17) results in

$$\begin{aligned}\frac{1}{2}\langle|\nabla\mathbf{u}|^2\rangle = -\frac{1}{2}\left\langle|\nabla\tilde{\mathbf{u}}|^2 + \frac{d\phi}{dz}\tilde{u}\tilde{w}\right\rangle +\\ + Gr(\overline{u}(1) - \phi(1)) + \frac{1}{2}\left\langle\left(\frac{d\phi}{dz}\right)^2\right\rangle,\end{aligned} \tag{19}$$

which allows us to define the quadratic expression

$$\mathcal{Q} = \left\langle |\nabla \tilde{\mathbf{u}}|^2 + \frac{d\phi}{dz} \tilde{u}\tilde{w} \right\rangle. \tag{20}$$

Using $\langle |\nabla \mathbf{u}|^2 \rangle = Gr\overline{u}(1)$, we can rearrange (19) into a more useful form

$$\overline{u}(1) = 2\phi(1) - \frac{1}{Gr}\left\langle \left(\frac{d\phi}{dz}\right)^2 \right\rangle + \frac{2}{Gr}\mathcal{Q}. \tag{21}$$

(21) is of great significance: given that we are free to choose the background field ϕ, if we choose one that ensures that $\mathcal{Q}$ is positive, we obtain a lower bound for $\overline{u}(1)$ that only depends on the background field of the form

$$\overline{u}(1) \geq 2\phi(1) - \frac{1}{Gr}\left\langle \left(\frac{d\phi}{dz}\right)^2 \right\rangle =: \mathcal{B}\{\phi\}, \tag{22}$$

which is the definition of the functional $\mathcal{B}$. $\mathcal{B}$ in this case provides a lower bound on $\overline{u}$.

By the definition of Gr and C_ε, we link C_ε to this new found bound such as

$$\begin{aligned} \varepsilon &:= \frac{\nu^3}{h^4}\langle \|\mathbf{u}\|^2 \rangle, \\ C_\varepsilon &:= \frac{\varepsilon h}{\overline{u_*}(h)^3} = \frac{Gr}{\overline{u}(1)^2}. \end{aligned} \tag{23}$$

Therefore, with the top non-dimensional velocity $\overline{u}$ and Gr one can obtain a value for C_ε. Using the value from the optimal background field yields a lower bound for the velocity, which in turns allows for an upper bound in C_ε.

In order to make the bound as precise as possible, we must choose a background field that maximises the functional $\mathcal{B}\{\phi\}$ whilst keeping $\mathcal{Q}$ positive. This will result in a tighter bound for C_ε. The method on how to obtain this optimal functional will be developed in the next sections.

Rescaling $\mathcal{Q}$ in Fourier space

The periodicity of the boundary conditions in x and y allow us to rescale $\mathcal{Q}$ in Fourier space. When doing this, it is also of interest to, for computational reasons [14], rescale z letting $\zeta = 2z - 1$ so that the vertical domain is in the interval $[-1, 1]$. Doing this, we rewrite $\mathbf{u}$ as

$$\mathbf{u}(x, y, \zeta) = \sum_{m,n\in\mathbb{Z}} \boldsymbol{U}_{mn}(\zeta) \mathrm{e}^{i(\alpha_m x + \beta_n y)}, \tag{24}$$

where

$$\alpha_m = \frac{2\pi m}{\Gamma_x}, \quad \beta_n = \frac{2\pi n}{\Gamma_y},$$

where Γ_i is the period in the ith axis, and $\boldsymbol{U}_{mn} = \mathbf{i}U_{mn} + \mathbf{j}V_{mn} + \mathbf{k}W_{mn}$. It should be noted that $\boldsymbol{U}_{mn}$ is a complex vector-valued function, and satisfies the incompressibility condition

$$i\alpha_m U_{mn} + i\beta_n V_{mn} + 2\frac{dW_{mn}}{d\zeta} = 0. \tag{25}$$

The Fourier-transformed boundary conditions are

$$\boldsymbol{U}_{mn}(-1) = \left.\frac{U_{mn}}{d\zeta}\right|_1 = \left.\frac{dV_{mn}}{d\zeta}\right|_1 = W_{mn}(1) = 0. \tag{26}$$

Inspired by the incompressibility equation, we introduce the rescaled and Fourier-transformed gradient-like operator

$$\mathcal{D} := \left(i\alpha_m, i\beta_n, 2\frac{d}{d\zeta}\right). \tag{27}$$

Using the expression for the Fourier velocity in (24) we can define a Fourier-transformed expression for $\mathcal{Q}$ such as

$$\mathcal{Q}_{mn} := \int_{-1}^{1}\left[\|\mathcal{D}\boldsymbol{U}_{mn}\|^2 + 4\frac{d\phi}{d\zeta}Re(U_{mn}W_{mn}^*)\right]d\zeta, \tag{28}$$

which allows us to write the spectral constraint as

$$\mathcal{Q}\{\mathbf{u}, \phi\} = \frac{1}{2}\mathcal{Q}_{00}\{\boldsymbol{U}_{00}, \phi\} + \sum_{n\geq 1}\mathcal{Q}_{0n}\{\boldsymbol{U}_{0n}, \phi\} + \sum_{m\geq 1}\sum_{n}\mathcal{Q}_{mn}\{\boldsymbol{U}_{mn}, \phi\}. \tag{29}$$

The spectral constraint must be satisfied by all flows, including those defined by a single pair of wavenumbers (α_m, β_n). Therefore, the constraint of $\mathcal{Q}$ being positive semidefinite is equivalent to require each $\mathcal{Q}_{mn}$ to be positive semidefinite.

Now that we have a rescaled, Fourier transformed expression of $\mathcal{Q}$, we can rescale the functional as well

$$\mathcal{B}\{\phi\} = \frac{2}{Gr}\int_{-1}^{1}\left(\frac{d\phi}{d\zeta}\right)^2 d\zeta - 2\phi(1), \tag{30}$$

where the difference with (22) (apart from the scaling) is that we have explicitly written the integral instead of leaving with spacetime average notation.

We now have rescaled expressions for the spectral constraint as well as the functional, which will allow us to set up a feasible problem to obtain bounds for C_ε in the following subsection. However, due to the nature of Fourier series, there can be an infinite number of wavenumbers in the solution. Wynn et al. [14] use basic functional estimates to prove that the only set of wavenumbers that should be considered for (20) is the one such that

$$\alpha_m^2 + \beta_n^2 \leq 2\left\|\frac{d\phi}{d\zeta}\right\|_\infty. \tag{31}$$

Therefore, we only need to consider the wave numbers up until m_{crit}, n_{crit} satisfying (31).

Obtaining a bound for C_ε in three dimensions

We now take the rescaled functional and spectral constraint and write an optimisation problem for the bound of C_ε for three dimensions. Equations 26 and 25 imply that $W_0 = 0$, meaning that $\mathcal{Q}_0 \geq 0$ for any choice of background field ϕ. Using the incompressibility condition

$$\nabla \cdot \mathbf{u} = \mathcal{D} \cdot \boldsymbol{U}_{mn} = i\alpha U_{mn} + i\beta V_{mn} + 2\frac{dW_{mn}}{d\zeta} = 0, \tag{32}$$

we obtain the relation $V_{mn} = -i\alpha U_{mn} - 2dW_{mn}/d\zeta$. Following [20], [7], we assume the system is translation invariant along the x axis, meaning $\alpha_m = 0$. This assumption greatly simplifies the algebra, as well as allowing for a simpler and less computationally expensive optimisation problem to implement. Although this assumption is unproven, Young et al. [20] found computationally this assumption to be true.

Substituting in (28), dropping m for simplicity and noting that

$$\begin{aligned}\|\mathcal{D}\boldsymbol{U}_n\|^2 &= -\beta^2\left(U_n^2 + V_n^2 + W_n^2\right) + 4\left(\left(\frac{dU_n}{d\zeta}\right)^2 + \left(\frac{dV_n}{d\zeta}\right)^2 + \left(\frac{dW_n}{d\zeta}\right)^2\right)\\ &= -\beta^2\left(U_n^2 - \frac{4}{\beta^2}\left(\frac{dW_n}{d\zeta}\right)^2 + W_n^2\right) + 4\left(\left(\frac{dU_n}{d\zeta}\right)^2 - \frac{16}{\beta^2}\left(\frac{d^2W_n}{d\zeta^2}\right)^2 + \left(\frac{dW_n}{d\zeta}\right)^2\right),\end{aligned} \tag{33}$$

we obtain the expression for $\mathcal{Q}$ as

$$\mathcal{Q}_n = \int_{-1}^{1}\left[\beta^2|U_n|^2 + 4\left|\frac{dU_n}{d\zeta}\right|^2 + \beta_n^2|W_n|^2 + 8\left|\frac{dW_n}{d\zeta}\right|^2 + \frac{16}{\beta^2}\left|\frac{d^2W_n}{d\zeta^2}\right|^2 + 4\frac{d\phi}{d\zeta}\mathrm{Re}(U_nW_n^*)\right]d\zeta. \tag{34}$$

Due to our previous assumption of x invariance, $\alpha_m = 0$. With this in mind, we can rearrange (31) to obtain the critical wavenumber to consider n_{crit} such as

$$n_{crit}(\phi) = \left\lfloor \frac{\Gamma_y}{\pi}\sqrt{\frac{1}{2}\left\|\frac{d\phi}{d\zeta}\right\|_\infty}\right\rfloor. \tag{35}$$

Only having to consider n up to a positive finite integer value greatly reduces the size of the problem .

Therefore, we can finally set up a optimisation problem to obtain the optimal background field

$$\begin{aligned} &\min_{\phi} \quad \mathcal{B}(\phi)\\ \text{s. t.}\quad &\mathcal{Q}_n\{U_n, W_n, \phi\} \geq 0,\ 1 \leq n \leq n_c(\phi),\\ &\phi(-1) = 0,\\ &\left.\frac{d\phi}{d\zeta}\right|_1 = \frac{Gr}{2}.\end{aligned} \tag{36}$$

As explained in subsection A.2, once the optimal background field has been obtained, one can compute the average velocity at $z = 1$ thanks to (22), and then proceed to obtain an upper bound for C_ε.

B. QUINOPT

QUINOPT [21] is an add on to the MATLAB optimisation toolbox YALMIP to solve optimisation problems with polynomial quadratic constraints, written by Fantuzzi, Wynn (Imperial College London), and Goulart and Papachristodoulou (University of Oxford). QUINOPT will be used to obtain optimal background fields for stress-driven shear flows at a range of Gr. The programme also computes the velocities that minimise $\mathcal{Q}_n$ with the appropriate modifications, which will be compared to DNS data.

The aim is to optimise an objective function, but QUINOPT does not include an optimisation solver. After comparing readily available solvers, SeDuMi was chosen. SeDuMi [22], an optimisation solver for symmetric cones, is recognised to be one to the best open source solvers for semidefinite programming, but as this is a relatively novel technique there are still improvements to be made: it is not rare for SeDuMi to run into numerical problems if the matrices in the problem are ill conditioned. This was the main limitation of running higher Gr optimisations, as high Gr leads to larger matrices that are more prone to ill conditioning.

The steps followed in the previous section allow for the spectral constraint to be only a function of U_n, W_n and ϕ only, as well as being expressed in Fourier space. Also, both $\mathcal{Q}$ and $\mathcal{B}$ are rescaled for better computational performance. However QUINOPT uses semidefinite programming to solve the optimisation problems, as explained in the Introduction. Therefore, the optimisation problem (36) is not suitable as it is. Wynn et al. [14] propose a semidefinite version of the optimisation problem such as

$$\begin{aligned}
\min_{\eta,\hat{\boldsymbol{\phi}},\boldsymbol{t}} \quad & \frac{2}{Gr}\eta - 4\phi_0 \\
\text{s. t.} \quad & \mathcal{S}(\hat{\boldsymbol{\phi}},\eta) \succeq 0, \\
& \mathcal{Q}_n(\hat{\boldsymbol{\phi}},\eta) \succeq 0, \qquad 1 \le n \le n_c, \\
& 1 - \kappa_n \mathbf{1}^T \boldsymbol{t} \ge 0, \qquad 1 \le n \le n_c, \\
& \phi_j - t_j \le 0, \qquad 0 \le j \le P, \\
& \phi_j + t_j \ge 0, \qquad 0 \le j \le P, \\
& \mathbf{1}^T \hat{\boldsymbol{\phi}} = \frac{Gr}{2}.
\end{aligned} \tag{37}$$

where $\hat{\boldsymbol{\phi}} = (\hat{\phi}_0, ..., \hat{\phi}_P)^T$ is the vector of Legendre coefficients, η is a slack variable (subject to $\eta \ge \hat{\boldsymbol{\phi}}^T \boldsymbol{B} \hat{\boldsymbol{\phi}}$), $\boldsymbol{t} = (t_0, ..., t_P)^T$ is a vector of slack variables, κ_m is a positive constant scaling as $\mathcal{O}(N^{-3})$, and $\boldsymbol{B}$ and $\mathcal{S}$ are defined as

$$\begin{aligned} \boldsymbol{B}_{rs} &= \frac{2\delta_{rs}}{2r+1}, \quad 0 \leq r, s \leq P, \\ \mathcal{S} &:= \begin{bmatrix} \boldsymbol{B}^{-1} & \hat{\boldsymbol{\phi}} \\ \hat{\boldsymbol{\phi}}^T & \eta \end{bmatrix}, \end{aligned} \tag{38}$$

with δ_{rs} being the Kroenecher delta. Slack variables, a concept from convex optimisation, are usually employed to replace an inequality constraint with an equality constraint and a nonnegativity constraint [23].

Due to the derivation of $\mathcal{Q}_n$, the results from QUINOPT will be in the non-dimensional form of (2). This applies to the optimised background field as well as the velocity minimisers.

Internally, QUINOPT represents the variables (in our case, ϕ, u and w) as Legendre polynomials with the aim of setting up a system of LMIs for the semidefinite programme. The non-periodic boundary condition makes a Fourier series infeasible, and the analysis requires orthogonality of the functions. This makes Legendre expansions ideal [14]. Moreover, the use of Legendre series makes the differentiation needed to evaluate $\mathcal{Q}_n$ trivial.

To obtain results, all that is needed is to run the function `quinopt()` with the corresponding inputs. The function requires as inputs a symbolic expression of the quadratic expression and the appropriate boundary conditions.

The calculations were performed on a laptop computer with a Intel i7 CPU 4980HQ 2.20GHz.

C. Direct Numerical Simulations

As discussed during the introduction, direct numerical simulations (DNS) solve the Navier-Stokes equation computationally. This provides a very accurate result (assuming the grid definition is fine enough, and that the domain is big enough to capture all the relevant phenomena). However, direct numerical simulations come at a great computational cost and are not feasible for many engineering applications.

We will use the results obtained from DNS to validate the calculated optimal bounds for the dissipation coefficient, as well as exploring the effect of different Gr on the flow. The code, Diablo, used was specifically written for channel-like flows driven by shear stresses, and can be found in [24]. Diablo, written in Fortran, has been validated for channel flows, so we expect to obtain accurate results for a converged solution. This, combined with the lack of experimental data for stress driven shear flows justifies the lack of a validation section for the DNS.

Like most codes written to perform DNS, Diablo uses a different version of the non-dimensional Navier-Stokes than the one in (2), this equation being

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p &= \frac{1}{Re}\Delta\mathbf{u}, \\ \nabla\cdot\mathbf{u} &= 0, \end{aligned} \tag{39}$$

with boundary conditions

$$\mathbf{u}|_{z=0}= 0, \quad w|_{z=1}= 0, \quad \left.\frac{dv}{dz}\right|_{z=1} = 0, \quad \dot{m} = const. = 0.5, \tag{40}$$

where the Reynolds number is based on the bulk velocity of the flow. The boundary conditions for (39) are slightly unusual: instead of prescribing a value for the forcing shear stress, the code let the mass flow rate to be constant and adjust the forcing shear stress to meet this condition. It follows a behaviour such as

$$\left.\frac{du}{dz}\right|_{z=1}(t) = \frac{Gr(t)}{Re}, \tag{41}$$

with Re being prescribed as $1/\nu$ and Gr changing with time to ensure the boundary condition is satisfied. For a converged solution, the Grashoff number will converge to a finite value. Compared with prescribing Gr, this approach leads to quicker convergence and a more efficient code. This equation is not explicitly written in the code as a function of Gr but as a function of Re_τ, but it has the same behaviour of allowing the shear driving parameter to change with time. The shear Reynolds number Re_τ is defined such as

$$Re_\tau = \frac{u_\tau L}{\nu}, \tag{42}$$

where the shear velocity u_τ is defined as τ_w/ρ.

(39) was obtained using the non-dimensional quantities

$$\boldsymbol{x} = \frac{\boldsymbol{x}_*}{h}, \quad t = \frac{t_* U_{ref}}{h}, \quad \mathbf{u} = \frac{\mathbf{u}_*}{U_{ref}}, \quad p = \frac{p_*}{\rho U_{ref}^2}, \tag{43}$$

A few modifications had to be included in Diablo to obtain all necessary data, as the DNS didn't directly output the instantaneous Fourier velocities (although it used them for some calculations).

In order to perform the velocity calculations, Diablo employs Fourier transforms of the velocities to obtain their coefficients $\boldsymbol{U}_n$, using a fast Fourier transform algorithm, where $\boldsymbol{U}_n = (U_n, V_n, W_n)$. This algorithm is optimised to work on a 2π periodic domain, meaning $\Gamma_x = \Gamma_y = 2\pi$. When reading the Fourier-transformed velocities from the variables in the code this is the obtainable value. In order to correct for this, we start from the definition of the discrete Fourier transform. Diablo defines de Fourier components to satisfy

$$\int_0^{2\pi} |v(y)|^2 dy = \sum_{n=-\mathrm{NK}_y}^{\mathrm{NK}_y} |V_n|^2. \tag{44}$$

The limits $[-\mathrm{NK}_y, +\mathrm{NK}_y]$ are set to be $\mathrm{NK}_y = \frac{1}{3}\mathrm{N}_y$, where N_y is the number of grid points in the spanwise direction. This result is obtained following the 2/3 rule [25] , which indicates that for numerical stability only 2/3 of the grid points should be considered as wavenumbers when performing the Fourier transform.

In order to correct for the mismatch in domain, we define new Fourier coefficients such as

$$\int_0^{\pi} |v(\hat{y})|^2 d\hat{y} = \sum_{n=-\mathrm{NK}_y}^{\mathrm{NK}_y} |V_n'|^2. \tag{45}$$

letting $\hat{y} = \frac{1}{2}y$, and substituting in the previous equation, we obtain

$$\begin{aligned} 2\int_0^{\pi} |v(\hat{y})|^2 d\hat{y} &= \int_0^{2\pi} |v(y)|^2 dy, \\ \therefore \int_0^{\pi} |v(\hat{y})|^2 d\hat{y} = \frac{1}{2}\sum_{n=-\mathrm{NK}_y}^{\mathrm{NK}_y} |V_n|^2 &= \sum_{n=-\mathrm{NK}_y}^{\mathrm{NK}_y} \left|\frac{V_n}{\sqrt{2}}\right|^2 = \sum_{n=-\mathrm{NK}_y}^{\mathrm{NK}_y} |V_n'|^2. \end{aligned} \tag{46}$$

where V_n' are the corrected coefficients that we will use in our calculations. Therefore, all that is needed is to divide the coeffients from Diablo by $\sqrt{2}$. As the streamwise domain is set to be 2π, no correction is required for U_n, the x Fourier components.

Another inclusion was a finite difference algorithm for the Fourier velocities. (34) requires the values of the first and second derivatives of the Fourier velocities. A second order accurate, central finite difference scheme was used for both the first and second derivative, of the form of

$$\begin{aligned} \frac{d\boldsymbol{U}_n(z)}{dz} &\approx \frac{\delta \boldsymbol{U}_n(z+1) + (h-\delta)\boldsymbol{U}_n(z) - h\boldsymbol{U}_n(z-1)}{2h\delta}, \\ \frac{d^2\boldsymbol{U}_n(z)}{dz^2} &\approx \frac{\delta \boldsymbol{U}_n(z+1) - (h+\delta)\boldsymbol{U}_n(z) + h\boldsymbol{U}_n(z-1)}{\delta h^2}, \end{aligned} \tag{47}$$

where h is the distance between the point z and $z+1$, δ is the distance between z and $z-1$ and N_z is the number of gridpoints in the vertical domain. The differenciation is implied to be done on each component of the vector of Fourier velocities $\boldsymbol{U}_n$.

For the boundaries, a first order accurate forwards and backwards finite difference scheme were used for the $z = 0$ and the $z = 1$ boundaries, respectively. The forward scheme is

$$\begin{aligned} \frac{d\boldsymbol{U}_n(0)}{dz} &\approx \frac{\boldsymbol{U}_n(1) - \boldsymbol{U}_n(0)}{h}, \\ \frac{d^2\boldsymbol{U}_n(0)}{dz^2} &\approx \frac{h\boldsymbol{U}_n(2) - (h+\delta)\boldsymbol{U}_n(1) + \delta\boldsymbol{U}_n(0)}{\delta h^2}, \end{aligned} \tag{48}$$

where h is the distance between 0 and 1, and δ is the distance between 1 and 2.

The backwards scheme used is the following

$$\begin{aligned} \frac{d\boldsymbol{U}_n(N_z)}{dz} &\approx \frac{f(N_z) - f(N_z-1)}{h}, \\ \frac{d^2\boldsymbol{U}_n(N_z)}{dz^2} &\approx \frac{\delta f(N_z) - (h+\delta)f(N_z-1) + hf(N_z-2)}{\delta h^2}, \end{aligned} \tag{49}$$

where h is the distance between the point N_z and $N_z - 1$, and δ is the distance between $N_z - 1$ and $N_z - 2$.

For the integration of $\mathcal{Q}_n$, a trapezoidal method was chosen. The reason for this decision was the ease of implementation of the trapezoidal method, its computational efficiency and the fact that it is second order accurate, matching the $\mathcal{O}^2$ accuracy of the finite differences. The implementation of the trapezoidal integration is

$$\int_0^1 \mathcal{P} dz \approx \sum_{i=0}^{Nz-1} \frac{\mathcal{P}(i+1) + \mathcal{P}(i)}{2h}, \tag{50}$$

where $\mathcal{P}$ is the integrand of the spectral constraint in (34) and h is the distance between the point i and the point $i+1$.

The last modification worth mentioning was the need to interpolate values of the vertical velocity: Diablo implements a staggered grid in the vertical direction in a way that cells are created with nodes in the midpoints of the sides, instead of on the vertices, as illustrated in 2. The main advantage of a staggered grid is that instead of calculating pressure values at the nodes depending on velocities at the nodes, the pressure is calculated at the centre of the cell depending on velocities at midpoints, yielding a more accurate solution [26]. The need to undo the staggering comes from wanting to calculate $\mathcal{Q}_n$ for DNS data, which needs to be integrated at discrete points.

All the mentioned modifications have been included in Appendix B.

Diablo was run on one of the computers of the Aeronautics Department at Imperial College London. The desktop computer featured a Intel Xeon CPU E5-2665 0 2.40GHz, which was operated remotely using a secure shell protocol (*ssh*).

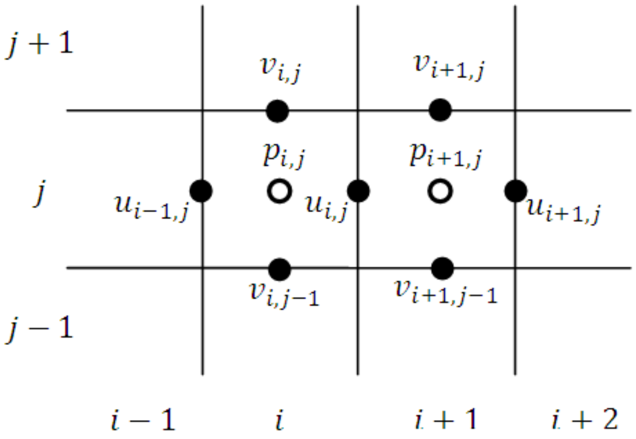

Figure 2: Staggered grid for CFD. Credit to S. Djeddi, A. Masoudi, P. Ghadimi [26].

D. Rescaling DNS results

The fundamental equations used by QUINOPT and DNS, although the same, are non-dimensionalised differently, as seen by inspecting equations (2) and (39). This means that the results from DNS are not directly comparable to the results from QUINOPT. By starting from one of the non-dimensional forms and obtaining the other one, we were able to derive relations for the non dimensional quantities such as

$$\boldsymbol{x}_{opt} = \boldsymbol{x}_{dns}, \quad t_{opt} = \frac{t_{dns}}{Re}, \quad \mathbf{u}_{opt} = Re\ \mathbf{u}_{dns}, \quad p_{opt} = Re^2\ p_{dns}. \tag{51}$$

The reason why modifications to the code were needed (as described in the previous section) is that the value of $\mathcal{Q}_n$ depends on the Fourier velocities. In fact, because of DNS scaling, equation (34) has to be rescaled before being implemented in the DNS. Using the chain rule, the equation resulted in

$$\mathcal{Q}_n = 2Re^2 \int_0^1 \left[\beta_n^2 |U_n|^2 + \left|\frac{dU_n}{d\theta}\right|^2 + \beta_n^2 |W_n|^2 + 2\left|\frac{dW_n}{d\theta}\right|^2 + \frac{1}{\beta^2}\left|\frac{d^2W_n}{d\theta^2}\right|^2 + 2\frac{d\phi}{d\zeta}\mathrm{Re}(U_n W_n^*) \right] d\theta, \tag{52}$$

where θ is the DNS domain, from 0 to 1 vertically. The term $d\phi/d\zeta$ is not changed as the values are directly read from the output from QUINOPT. Instead, a rescaling factor of 2 is applied. This equation was obtained using the relations

$$\begin{aligned} \boldsymbol{U}_{n\ opt} &= Re \boldsymbol{U}_{n\ dns}, \\ \frac{d\boldsymbol{U}_{n\ opt}}{d\zeta} &= Re \frac{d\boldsymbol{U}_{n\ dns}}{d\zeta} = Re \frac{d\theta}{d\zeta}\frac{d\boldsymbol{U}_{n\ dns}}{d\theta}, \\ \frac{d^2\boldsymbol{U}_{n\ opt}}{d\zeta^2} &= Re \left(\frac{d\theta}{d\zeta}\right)^2 \frac{d^2\boldsymbol{U}_{n\ dns}}{d\theta^2}, \\ \frac{d\theta}{d\zeta} &= \frac{1}{2}. \end{aligned} \tag{53}$$

In order to compare the results with those from [14], the following relations were also derived by substituting the previously derived relations in (51) into the definition of Gr and C_ε:

$$\begin{aligned} Gr = \frac{\tau_w h^2}{\rho \nu^2} = \frac{u_\tau^2 h^2}{\nu^2} = Re_\tau^2, \\ C_\varepsilon = \frac{\varepsilon h}{\overline{u_*}(h)^3} = \frac{Gr}{\overline{u}_{opt}(1)^2} = \frac{Gr}{(Re\ \overline{u}_{dns}(1))^2}. \end{aligned} \tag{54}$$

III. Results and Discussion

A. Results from DNS

In this section we explore the results from DNS for one specific simulation: $Re_\tau \approx 80$. The aim is to obtain some physical insight on how this type of flows behave. The reason for only studying one simulations is the relatively small change between the Re_τ range that we were able to obtain. This section is accompanied by Appendix A, where the same plots that are explained here can be found for $62 < Re_\tau < 143$.

The reason for not using round numbers for Re_τ is that Diablo does not allow for a direct input of Re_τ. As explained before, it has as inputs the Reynolds number Re, and it then tries to satisfy the constraint of constant mass flow rate by changing Re_τ. Therefore it was necessary to try different Re until the Re_τ desired was approximately obtained.

One of the challenges of this problem is to make sure the box size is large enough to capture all the relevant phenomena of the flow. We expect this type of flow to be similar to a Couette flow, and in the three dimensional case to present Langmuir-circulation-like flows [7]. This leads to spanwise vortices forming and is the source of the periodicity in the spanwise direction. The box size has to be large enough to capture this and other phenomena: it was chosen to be $\Gamma_x = 2\pi, \Gamma_y = \pi$ and $z = 1$ based on previous studies on Couette flow.

More importantly, the grid size needs to be defined. We based our first estimate for a grid size on previous DNS ran for Couette flow, giving a number of grid points of 65 in the z direction and 48 in the x and y direction. We then increased the resolution by approximately 50% ($N_x = 72$, $\mathrm{N}_y = 72$ and $N_z = 97$) and the velocity plots did not change noticeably. As the solution did not significant improve with an increase in resolution, it is safe to assume that the original resolution was sufficient. Nevertheless, as all the simulations had to be run at the higher resolution for verification, the results plotted are from these higher resolution simulations.

The results are as expected. For the case of $Re_\tau \approx 80$, the streamwise velocity profile is smooth and presents two boundary layers and an area near the center of the flow where the velocity gradient is smaller, as seen in Figure 3. The gradient of the velocity at the top layer is governed by the driving shear stress, with higher stresses leading to steeper boundary layers

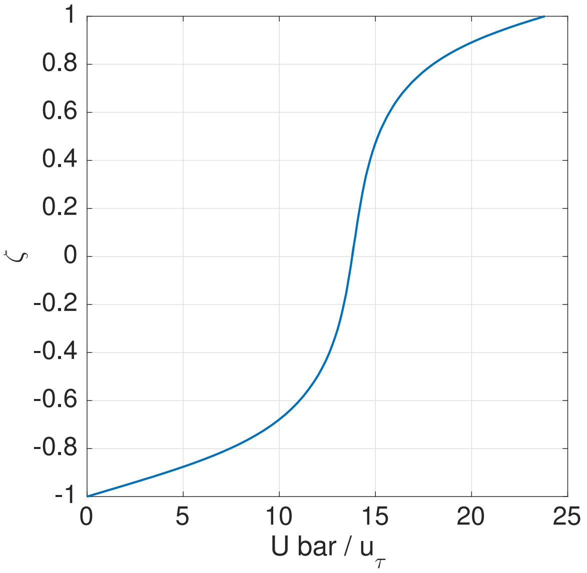

Figure 3: Plot of the average velocity profile along the vertical domain, normalised by u_τ, for $u_\tau = 3.73 \times 10^{-2}$ and $Re_\tau = 83.5$.

and the area near $z = 0.5$ becoming an area of constant flow velocity.

The root mean square values of the velocity, seen in Figure 4, represent the strength of the turbulence as the standard deviation of the fluctuations of the flow. The peak in the streamwise component u_{rms} occurs near $z = 0.15$, as the boundary layer produced by the no slip condition on the bottom surface induces shear and therefore turbulence. The strength of the turbulence is diminished as the boundary layer effects become smaller, and they increase again near the top surface. This is due to the applied shear stress producing an increase in turbulence near the top layer. It is noticeable that the turbulence does not go to zero on the top surface, as the no slip boundary condition is not being enforced.

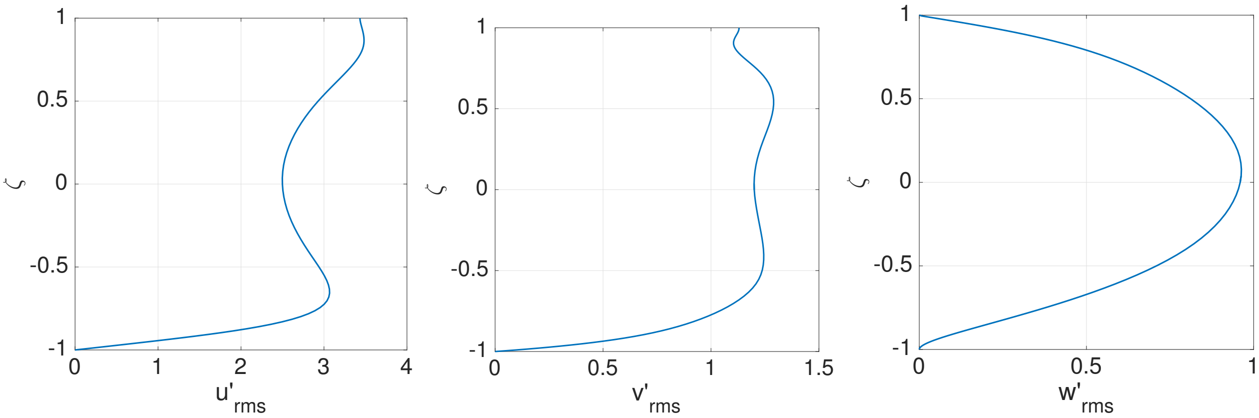

Figure 4: Plot of the root mean square velocity profiles along the vertical domain, for $u_\tau = 3.73 \times 10^{-2}$ and $Re_\tau = 83.5$. All velocities are normalised by u_τ.

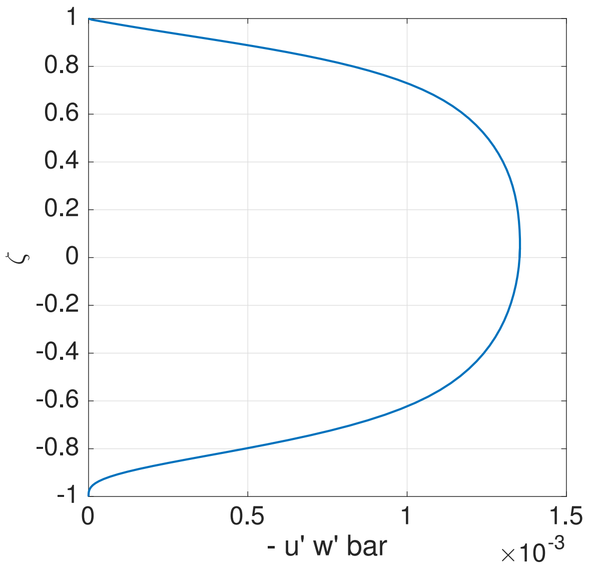

Figure 5: Plot of the time averaged Reynold stress $\overline{u'w'}$ along the vertical domain, for $u_\tau = 3.73 \times 10^{-2}$ and $Re_\tau = 83.5$.

The spanwise component v_{rms} has a very similar behaviour to u_{rms} near $z = 0$. However, instead of the turbulence reducing in the bulk of the flow, the strength remains fairly constant, and then presents the opposite pattern that u_{rms} near the top layer: a decrease in strength followed by an increase near $z = 1$. This behaviour can be attributed to the Langmuir circulation causing instabilities near the top layer in the spanwise direction.

The vertical component w_{rms} behaves smoothly and as expected: the increase of absolute velocity near the bulk of the flow leads to an increase in turbulence strength, and the boundary layer effects that dominated u_{rms} are substantially smaller in the z direction due to the non-permeability condition of both boundaries.

It is also of interest to discuss the Reynold stresses in the flow. The Reynold stresses can be physically understood as flux of the i momentum in the j direction, where i and j are different direction vectors. For example, $\rho\overline{uw}$ can be thought of as the turbulent vertical advection of streamwise turbulent momentum, or simply the vertical flux of streamwise momentum. Therefore these Reynold stresses provide an insight into the mixing properties of the turbulent flow.

With this in mind we begin our analysis. Looking at Figure 5, we see that the bulk of the flow reaches a value of $\overline{u'w'} \approx 1.4 \times 10^{-3}$, with the negative correlation of the fluctuations implying a positive value of $\partial\overline{u}/\partial z$, as expected. The value of $\overline{u'w'}$ tends to zero at both boundaries, due to the no slip condition in the bottom surface and the boundary condition $w'|_{z=1}= 0$. The fact that both u' and w' are required to be zero at the bottom surface, compared to the top surface where only w' is required to be zero is what explains the sharper gradient in the bottom surface.

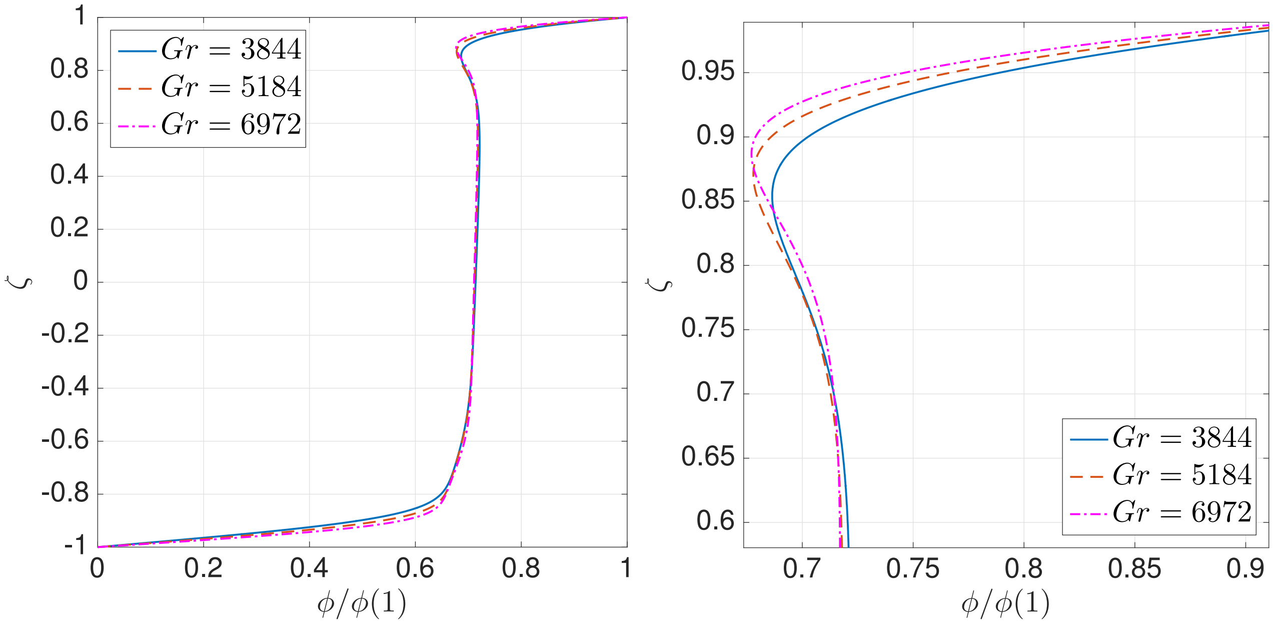

Figure 6: Normalised background field $\phi/\phi(1)$ for a range of Gr. The plot to the right emphasises the behaviour near the top boundary layer.

B. Results from QUINOPT

Using the techniques explained in previous sections, the SDP was solved for a range of Gr between ≈ 3800 and ≈ 12800. The specific values for Gr were chosen to match the Re_τ^2 of the DNS results, and hence the lack of roundness in the values obtained. To be able to compare the results, the periodic domain chosen was $\Gamma_y = 3.14$.

The implementation consisted of estimating a critical wavenumber, then calculating n_{crit} thanks to (35) and then rerun QUINOPT if the critical wavenumber was larger than the initial guess. We followed Wynn et al. [14] for our estimates for n_{crit} and the Legendre polynomial P.

Looking at Figures 6 and 7, it is easy to observe the behaviour of ϕ with Gr. As discussed in Wynn et al, the background field behaves linearly at very low Gr, in the range of $0 < Gr \leq 50$. It quickly starts developing two boundary layers, near $\zeta = -1$ and $\zeta = 1$, as well as a profile in between them that flattens out with increased Gr.

As Gr is increased, the bottom boundary layer develops two distinct regions: a region of high gradients between $-1 < \zeta < -0.85$ and another one with a flatter profile for $-0.85 < \zeta < -0.55$. The top boundary layer also develops two regions, one with a negative gradient between $0.8 < \zeta < 0.9$ and another one mirroring the high gradient region in the bottom boundary layer for $\zeta > 0.9$.

Overall, all these features are exacerbated as Gr is increased, meaning a correlation with high gradients and a flatter profile with increased driving parameter Gr.

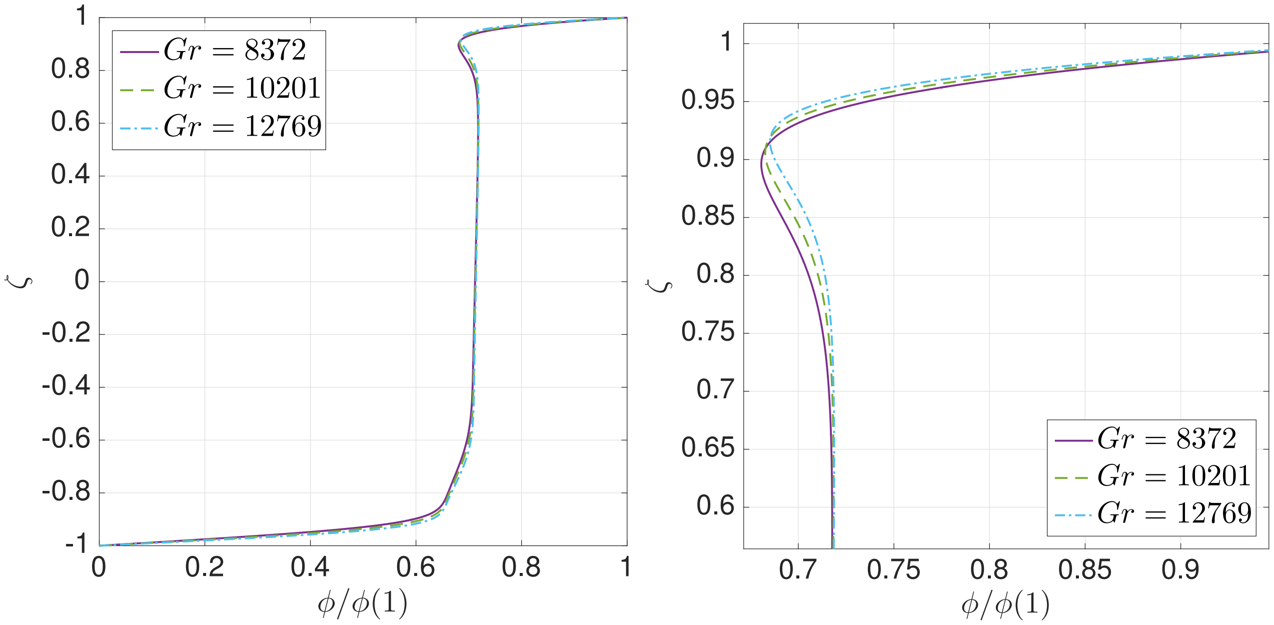

Figure 7: Normalised background field $\phi/\phi(1)$ for a range of Gr. The plot to the right emphasises the behaviour near the top boundary layer.

The formulation of QUINOPT did not allows us to directly obtain the velocity minimisers U_{n0}, W_{n0}, that optimise the functional $\mathcal{B}$, or the values of the expression Q_n. Instead, we solved the problem as usual, obtaining the expression for the background field, and then modified the spectral constraint such as

$$\mathcal{Y}_n = \mathcal{Q}_n - \lambda_0(\|U_n\|^2 + \|W_n\|^2), \tag{55}$$

and then let QUINOPT optimise λ_0, obtaining its value as a function of the wavenumber. λ_0 is defined such as

$$\lambda_0 = \frac{Q_n}{\|U_{n0}\|^2 + \|W_{n0}\|^2}. \tag{56}$$

The behaviour of λ_0 with increasing wavenumber is of great interest. We will focus for now on the case $Gr \approx 3800$. As seen in Figure 8, the positivity of λ_0 is ensured at high wavenumbers. However, certain wavenumbers present a value of λ_0 very close to zero, namely $n = 2,\ 5,\ 6$. These wavenumbers will be referred as the *critical wavenumbers*, and they are fundamental to the solution as they are the waveumbers that can threaten the condition of the spectral constraint being strictly positive.

Figure 9 follows the previous analysis and extends it to a range of Gr. The first noticeable feature is that as Gr is increased, the plots seem to be shifted to the right (as well as slightly changing gradients), meaning an increase in Gr leads to more critical wavenumbers. This is in accordance with our previous analysis and condensed in (35). More importantly, when looking at the zoomed-in plot, the critical wavenumbers of λ_0 seem to shift with increased Gr, losing some of the lower critical wavenumbers and trading them for larger critical wavenumbers. Table

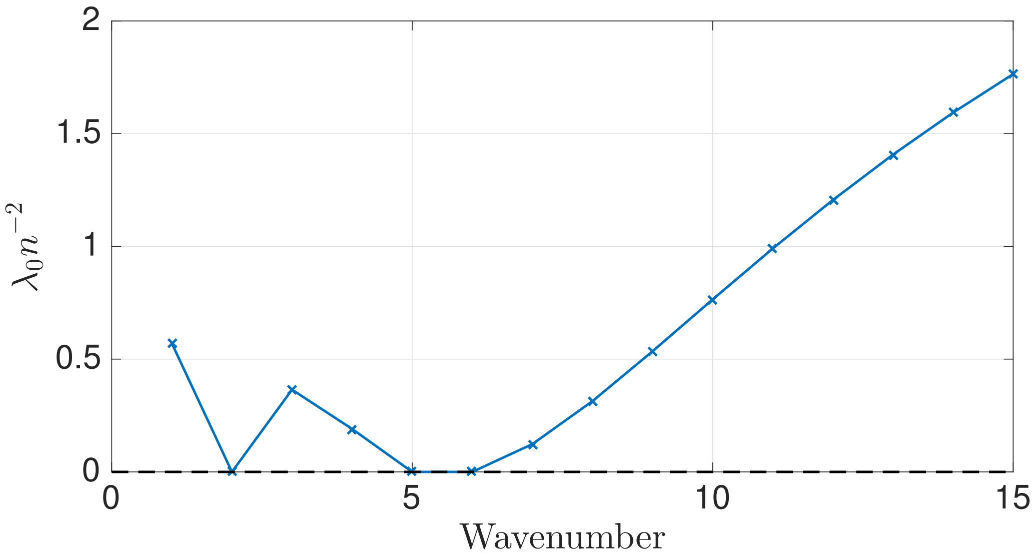

Figure 8: Evolution of λ_0 with wavenumber for $Gr = 3844$. Notice the critical wavenumbers at $n = 2, 5, 6$.

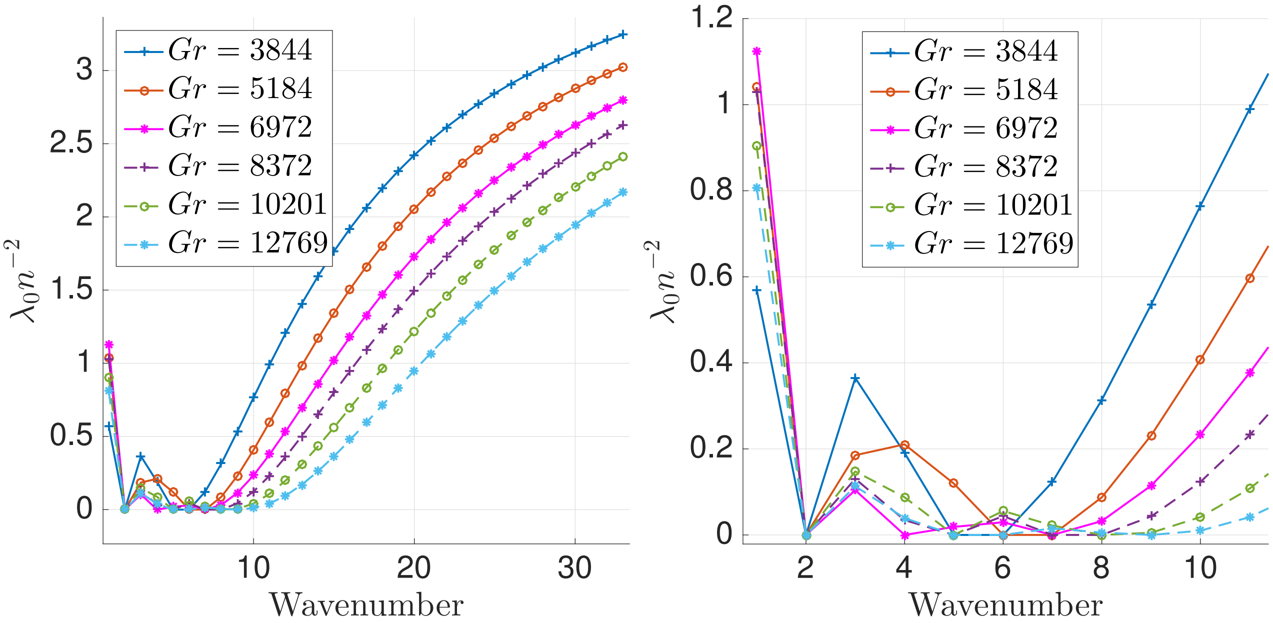

Figure 9: Evolution of λ_0 with wavenumber for a range of Gr. The plot to the right emphasises the behaviour at low wavenumbers.

Table 1: Critical wavenumbers for a range of Gr.

Gr	n=1	2	3	4	5	6	7	8	9	10
3844		✓			✓	✓				
5184		✓				✓				
6972		✓		✓			✓			
8372		✓			✓		✓			
10201		✓			✓			✓	✓	
12769		✓							✓	✓

1 shows the critical wavenumbers for the chosen range of Gr, and the transition to having a third bifurcation is even clearer. This third bifurcation is associated with the internal structure in the bottom boundary layer discussed before [14], with the two regions only developing after the third bifurcation occurs.

Nicodemus et al. [4] employed the background field method for plane Couette flow, and obtained results for a wide range of Re. In this case, the driving parameter is not Gr as the boundary condition is not one of shear stress. When compared to the results we present, the top boundary layer does not feature the overshoot region with negative derivative, but a sharp change directly to the second region discussed before. Regarding the bottom region, the plane Couette formulation does have the characteristic two sharp increased followed by a more gradual increase in ϕ before the flat central region. This indicates that the behaviour of the optimised ϕ, at least near the bottom boundary layer, does not depend on the surface forcing, but on the behaviour of the critical wavenumbers.

C. Behaviour of $\mathcal{Q}_n$

We now turn to investigate and discuss the behaviour of the spectral constraint $\mathcal{Q}_n$. As discussed in the Mathematical Framework subsection, C_ε depends on the mean value of the velocity at the top layer. This velocity itself is bounded by the functional $\mathcal{B}$. This lower bound on the velocity is a function of the background field, meaning that a more accurate background field will provide a better bound.

However, the methods explored are already considered optimal, while their results are an order of magnitude larger than the experimental data suggest [14]. Therefore, instead of exploring a seemingly already exhausted variational principle, we will compare it to the values of $\mathcal{Q}_n$ obtained from DNS.

In order to obtain values for $\mathcal{Q}_n$ from Diablo, a background field is required first. Therefore, we ran QUINOPT at the required Gr and then used linear interpolation to adjust it to the number of gridpoints in the DNS. Once this was done, a simple trapezoidal method as explained in section C was implemented.

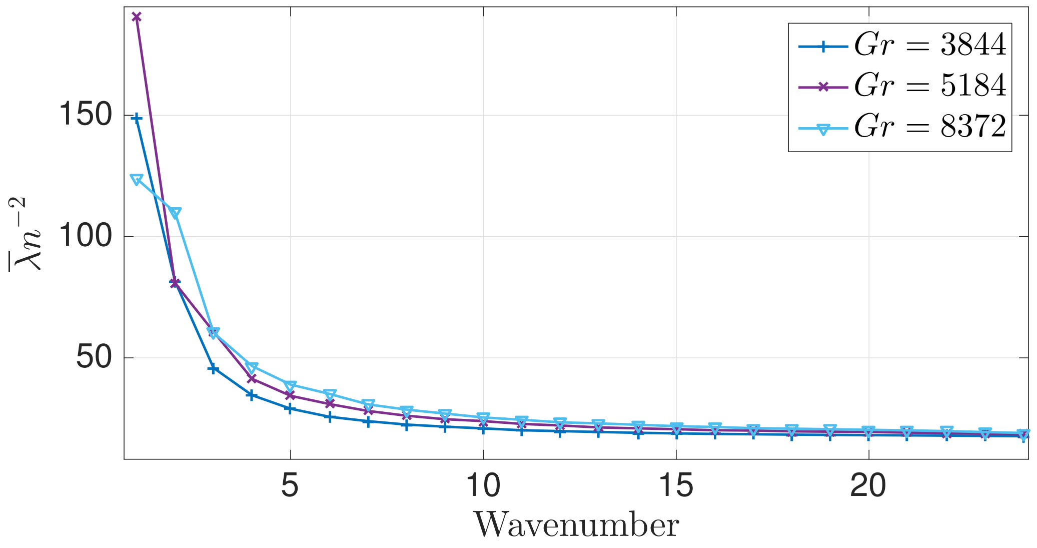

Figure 10: Evolution of $\overline{\lambda}_n$ with wavenumber for a range of Gr.

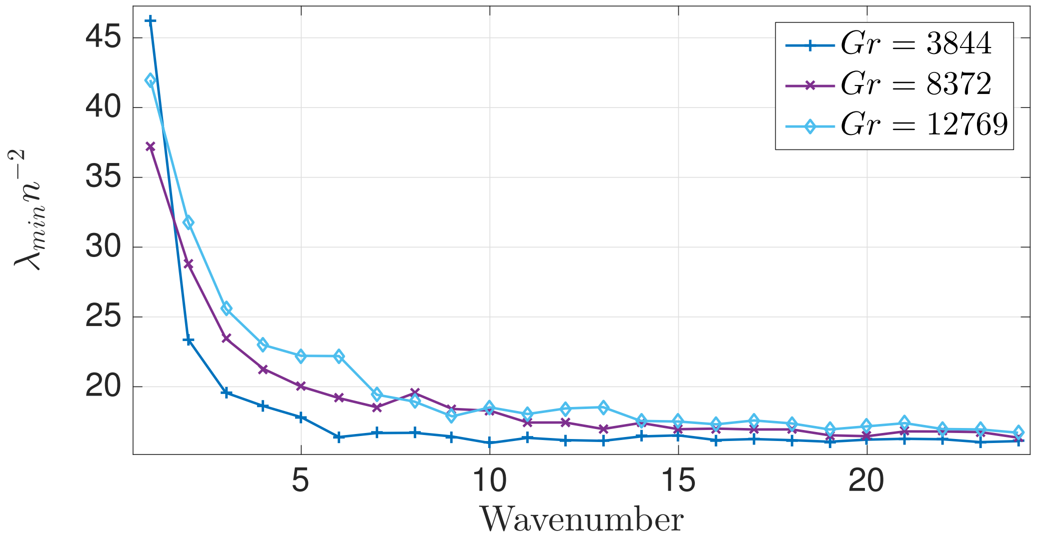

Figure 11: Evolution with wavenumber of λ_{min} for a range of Gr.

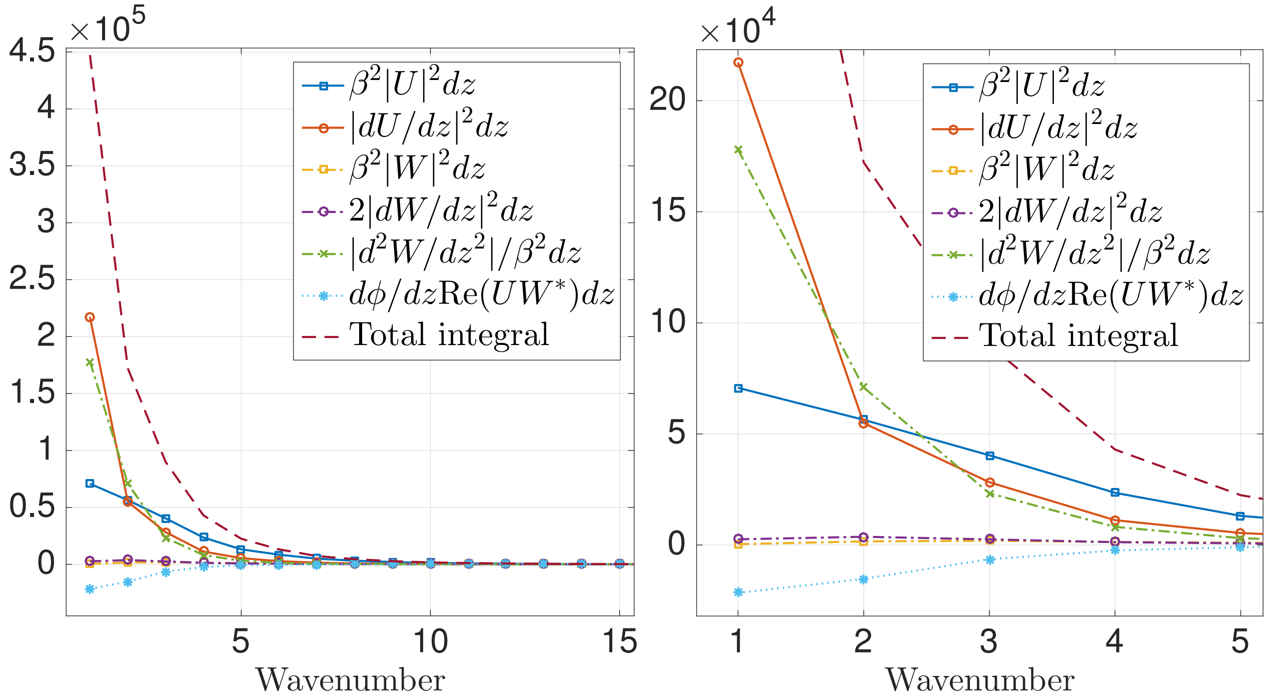

Figure 12: Evolution of each term of $\mathcal{Q}_n$ from DNS, for $Gr = 3844$. All quantities are integrated in the DNS domain $[-0.5, 0.5]$. The plot to the right emphasises the behaviour at low wavenumbers. All quantities are time averaged.

Each term of the integrand of $\mathcal{Q}_n$ was tracked and integrated individually, allowing for visualisation of the different terms. Figure 12 shows that $\mathcal{Q}_n$ decreases monotonically with wavenumber, with the background field term $d\phi/d\zeta \mathrm{Re}(U_n W_n^*)$ being the only negative term. The positivity of $\mathcal{Q}_n$ is dominated by the first derivative of U_n and the second of W_n, with $\beta^2|U_n|^2$ providing a smaller contribution at lower wavenumbers. After the third wavenumber, this term drives the positivity thanks to its β^2 factor, as opposed to the $1/\beta^2$ of the second derivative of W_n which greatly reduces its contribution.

Motivated by this analysis, and wanting to compare these results to those obtained through the optimisation problem, we introduce two quantities:

$$\overline{\lambda} = \frac{\overline{\mathcal{Q}_n}}{(\overline{\|U_n\|^2} + \overline{\|W_n\|^2})}, \quad \lambda_{min} = min\left(\frac{\mathcal{Q}_n}{(\|U_n\|^2 + \|W_n\|^2)}\right) \tag{57}$$

where min implies the minimum value that λ_n takes for each wavenumber, and $\overline{\mathcal{Q}_n}$ is the time average of the spectral constraint.

Figures 10 and 11 show the change with wavenumber of $\overline{\lambda}$ and λ_{min}, respectively. They were plotted with a factor of n^{-2} for the sake of clarity, as these quantities increase rapidly with wavenumber. Both appear to increase with Gr, although the behaviour of λ_{min} is somewhat more uncertain as it presents a less smooth pattern.

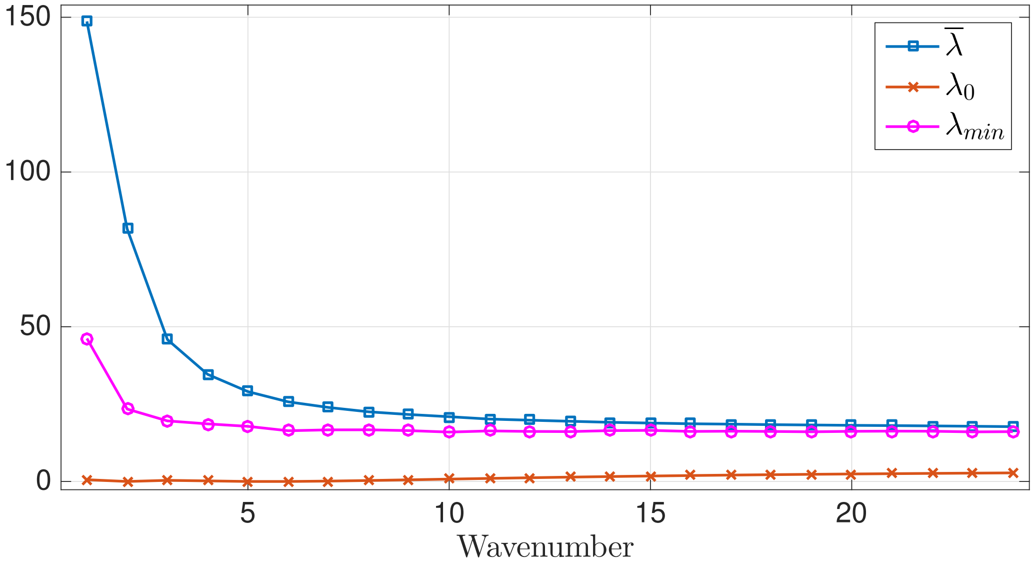

Figure 13: Evolution with wavenumber of λ_0 from QUINOPT, λ_{min} and $\overline{\lambda}$ from DNS, for $Gr = 3844$. All λ are multiplied by n^{-2}.

The value of $\overline{\lambda}$ is higher at low wavenumbers than λ_{min} by approximately a factor of two, but tends to a similar value after the $10th$ wavenumber. However, the high values at low wavenumbers for $\overline{\lambda}$, λ_{min} are in sharp contrast to the behaviour of λ_0. Figure 13 shows the comparison between the different quantities discussed for the case $Gr \approx 3800$. The main difference between the DNS results and the optimisation results is the strong positivity at low wavenumbers for the DNS results. While the DNS presents monotonically decreasing and stricly positive values, the positivity of λ_0 is threatened at low wavenumbers, and its value actually increases with wavenumber.

When analysing $\lambda 0$ at larger wavenumbers, it seems to approach this asymptotic value between 15 and 20. However, not enough data was collected to confirm this behaviour.

A clear difference between both the behaviour and the value of DNS and QUINOPT results is apparent in Figure 13. The culprit of this difference can only be the formulation: when optimising the background field, QUINOPT searches for the velocity minimisers, giving $\mathcal{Q}_n$ a value of zero (or almost zero) at the critical wavenumbers. However, these velocity minimisers do not necessarily occur in real flows, imposing an overly harsh contraint in the problem. This non-physicality of the mathematically feasible solution is what motivates the next section: the introduction of a correction factor in the formulation to try to restrict the background field to physical flow fields.

D. Correction factor ξ_n

With the previous section as motivation, we decide to introduce two correction factors such as

$$\xi_n^{min}(Gr) = \lambda_{min} - \lambda_0, \qquad \overline{\xi}_n(Gr) = \overline{\lambda} - \lambda_0, \tag{58}$$

which will in turn change the spectral constraint, giving us the modified spectral constraint

$$\overline{\mathcal{Q}}_n' = \mathcal{Q}_n + \int_{-1}^{1} \overline{\xi}_n(\|U_n\|^2 + \|W_n\|^2), \quad \mathcal{Q}_n'^{min} = \mathcal{Q}_n + \int_{-1}^{1} \xi_n^{min}(\|U_n\|^2 + \|W_n\|^2). \tag{59}$$

It is important to remember that these spectral constraint don't have the rigorous mathematical background that $\mathcal{Q}_n$ does, and therefore they do not have a physical meaning: they are an attempt at *relaxing* the problem. By adding these strictly positive terms to $\mathcal{Q}_n$, its positivity is more easily ensured, relaxing the requirements in the optimisation problem and yielding modified background fields $\phi_{\overline{\xi}_n}$ and $\phi_{\xi_n^{min}}$. The justification on why these correction factors improve the solution will be explored in the following sections.

E. Scaling of ξ_n

Looking at Figure 13, it is easy to see that the value of the correction factor will be a function of the wavenumber. Figure 14 illustrates this point for $\overline{\xi}_n$: the factor does increase with wavenumber, as well as with Gr. Based on these plots, and thanks to the curve fitting suite `cftool` for MATLAB, a power fit was used to fit the data with the aim of producing scaling laws. These fits can be seen for a range of Gr in Table 2. As well as presenting fits for different Gr, the table also shows values for typical error measurement indicators in fits. In simple terms, R-squared measures how much of the variance of the data is explained by the model or fit, and adjusted R-squared takes also the number of independent variables into consideration. RMSE, or root mean square error, measures the difference between the predicted and the obtained values. Therefore, a lower RMSE is correlated with better fits, and the closer R-squared and the adjusted R-squared are to unity, the better the model (assuming overfitting is not an issue).

There is a clear pattern of decreasing exponent and increasing multiplying constant with increasing Gr. This motivated us to obtain an equation that takes this Gr dependence into consideration, obtaining

$$\overline{\xi}_n(Gr) = 1.352 \times Gr^{0.4243} \times n^{(-1.8352\times10^{-5}Gr+1.824)}. \tag{60}$$

Looking at Table 3, no similar Gr dependence can be found for ξ_n^{min}. It could have been argued that a potential choice of ξ_n^{min} of the form of a power fit as a function of n that simply takes the average of all values with Gr would suffice. However, this will not yield better results than $\overline{\xi}_n$, as ξ_n^{min} is always smaller than $\overline{\xi}_n$. Therefore, as long as $\overline{\xi}_n$ provides a valid modified ϕ, it is not necessary to consider ξ_n^{min} as an option.

F. Justification of ξ_n: effect on background field ϕ

The implementation of the correction factor was relatively straightforward. Using the previously derived expression for $\overline{\xi}_n$, we were able to calculate the modified spectral constraint $\overline{\mathcal{Q}}_n'$

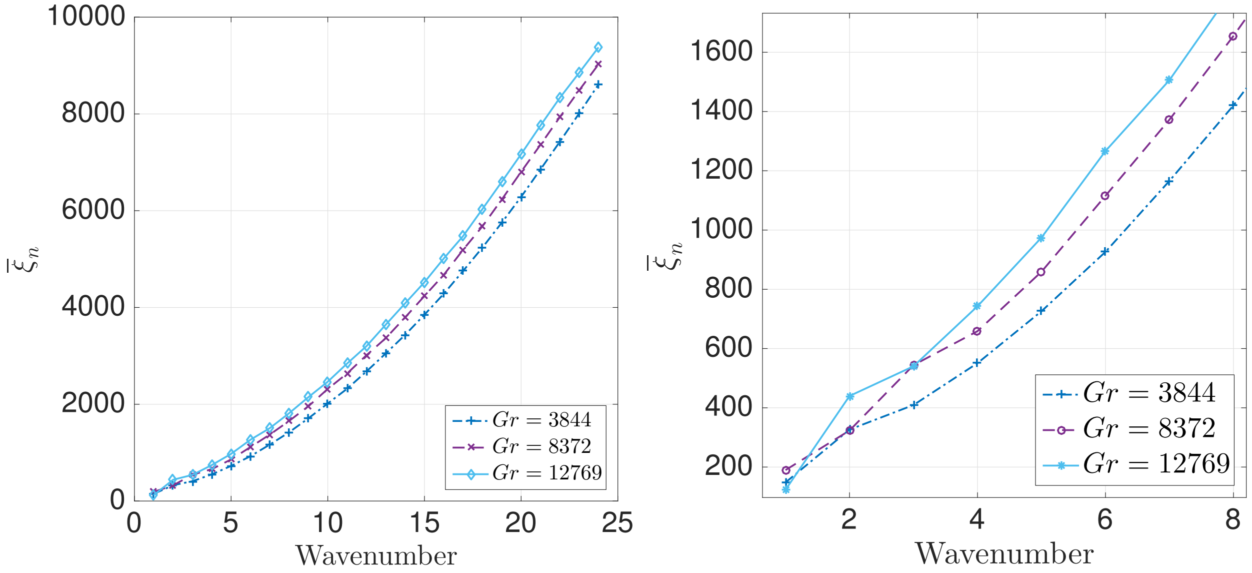

Figure 14: Evolution of $\overline{\xi}_n$ with wavenumber. The plot on the right emphasises the behaviour at low wavenumbers.

Table 2: Fits for $\overline{\xi}_n$

Gr	Equation	R-squared	adjR-squared	RMSE
3844	$43.53n^{1.661}$	0.9992	0.9991	77.72
5184	$52.27n^{1.61}$	0.9989	0.9988	93.26
6972	$59.33n^{1.576}$	0.9987	0.9986	103
8372	$61.94n^{1.567}$	0.9991	0.9991	55.46
10201	$65.85n^{1.554}$	9.9995	0.9994	68.52
12769	$75.61n^{1.518}$	0.9991	0.9991	89.48

Table 3: Fits for ξ_n^{min}

Gr	Equation	R-squared	adjR-squared	RMSE
3844	$32.75n^{1733}$	0.9995	0.9995	57.16
5184	$24.71n^{1.81}$	0.9998	0.9997	38.94
6972	$30.7n^{1.748}$	0.9998	0.9998	36.54
8372	$29.68n^{1.757}$	0.9995	0.9995	57.28
10201	$27.87n^{1.782}$	9.9995	0.9994	60.15
12769	$31.75n^{1.733}$	0.9995	0.9995	57.19

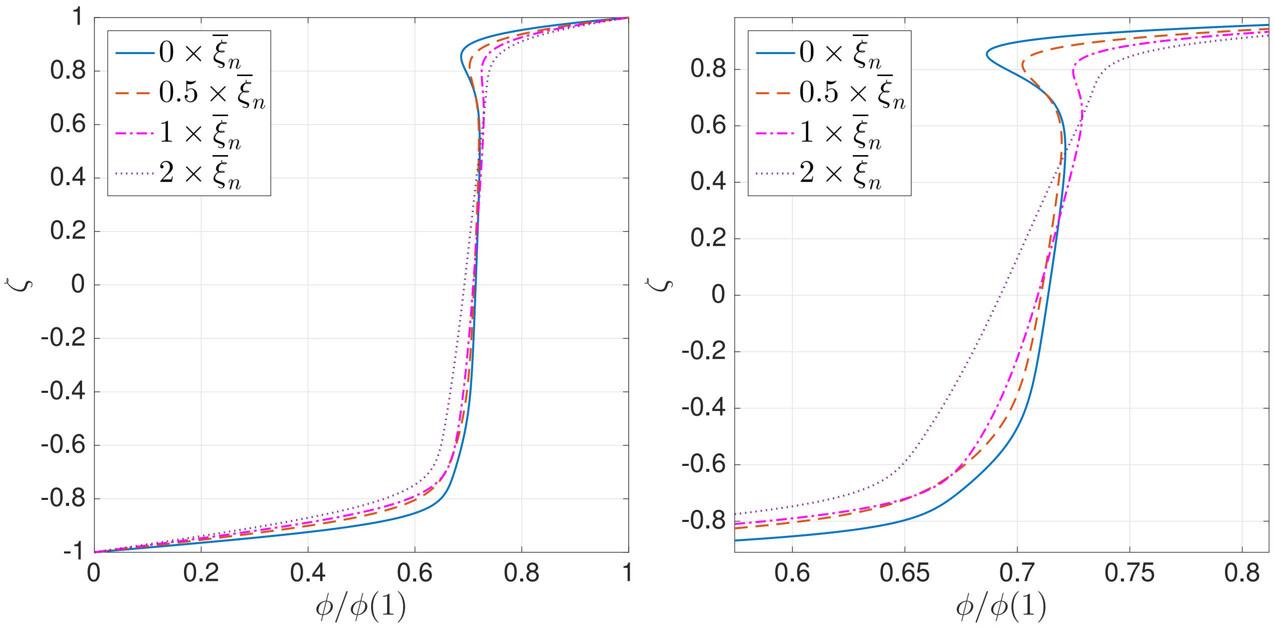

Figure 15: Changes of the normalised background field $\phi/\phi(1)$ with $\overline{\xi}_n$ for $Gr = 3844$. The plot to the right emphasises the behaviour of the top and bottom bottom boundary layers.

and implement it in QUINOPT. QUINOPT then returned the values for the modified background field $\phi_{\overline{\xi}}$ and the corrected normalised spectral constraint λ_0'.

Firstly we explore what effect the correction factor has on the background field. Figure 15 shows the background fields for different values of the correction factor $\overline{\xi}_n$. Increasing the correction, in this formulation, is essentially increasing the energy of the flow by increasing the $\|U_n\|^2+\|W_n\|^2$ terms. This leads to an overall increase in the value of ϕ, with a considerably smaller increase in the absolute value of the derivative with respect to the vertical domain. However, the normalised gradient is reduced. This point is illustrated in Figure 16. This is why using $\overline{\xi}_n$ is of great interest: adding the correction factor makes the background field smoother, and qualitatively makes it behave like a lower Gr background field, while the absolute value of the background field increases, which is something that occurs with increasing Gr.

This effect is shown in Figure 17. It can be concluded that adding the correction factor does indeed make the background field more *physical*, in the sense that it brings it closer to the values obtained for $\overline{u}$ from DNS for a give Gr.

G. Validation of ξ_n

The previous section provided a plausible explanation on why the scaling factor makes physical sense. However, a proper validation is still needed. The modified background field $\phi_{\overline{\xi}}$ must still satisfy the spectral constraint in the DNS.

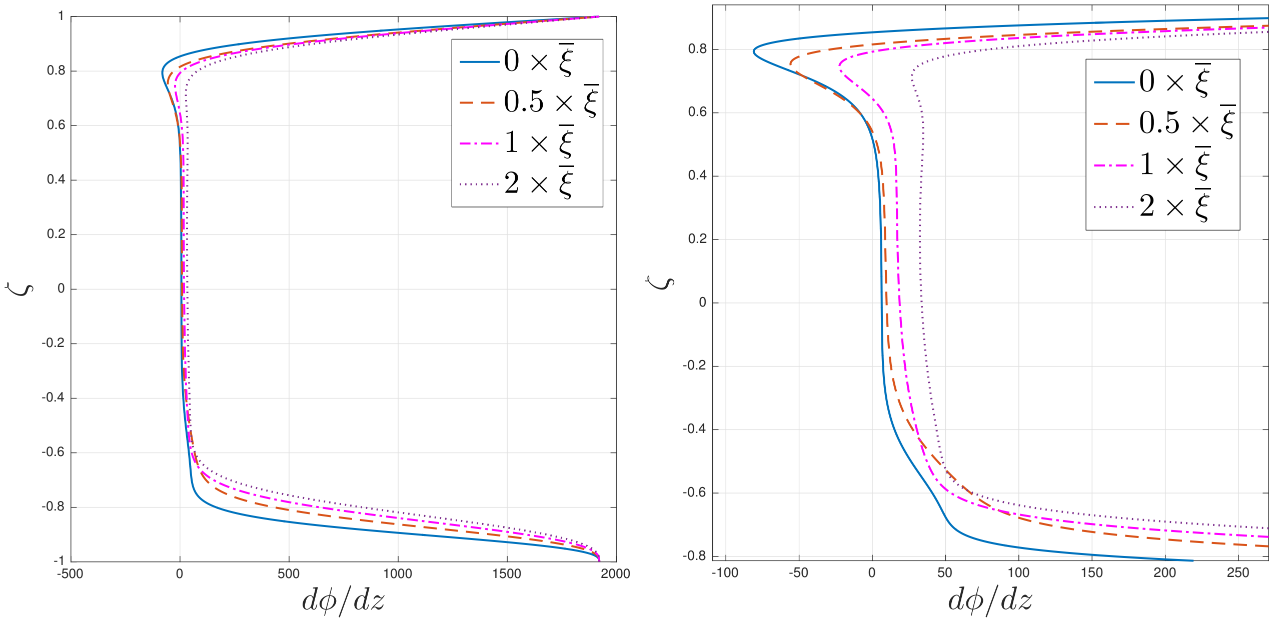

Figure 16: Changes of the derivative of background field $d\phi/d\zeta$ with $\overline{\xi}_n$. The plot to the right emphasises the behaviour of the top and bottom bottom boundary layers.

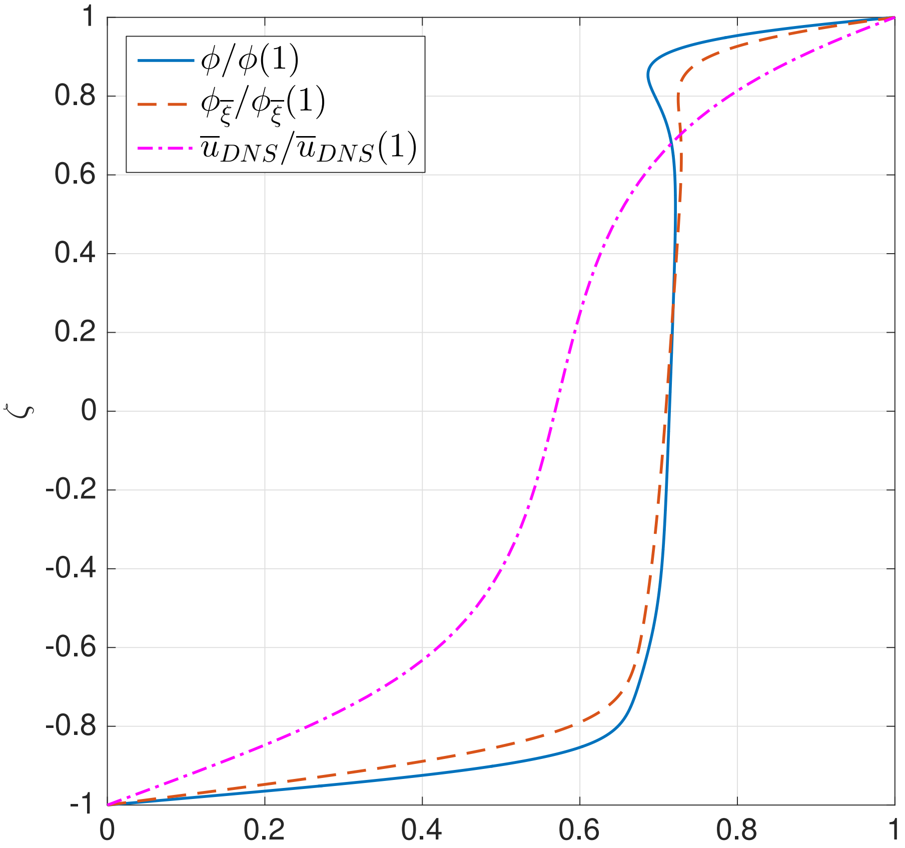

Figure 17: Comparison between the normalised background field, the normalised corrected background field and the normalised horizontal mean velocity from DNS, for $Gr = 3844$.

As the correction factor is essentially a relaxation of the problem, we might expect the corrected ϕ to struggle to satisfy the constraint without the help of the correction factor bringing positivity. Due to the formulation of $\mathcal{Q}_n$, the only term driving the negativity of the spectral constraint is the Fourier Reynold stress $\mathrm{Re}(U_n W_n^*)$ multiplied by the derivative of the background field.

However, we found no evidence that this is the case. Figure 18 contains the DNS information for the modified background field, and the overall expression is actually more positive than the original expression shown in Figure 12. As nothing else has been modified, this positivity must be a direct consequence of the addition of the correction factor $\overline{\xi}_n$.

As discussed previously, adding $\overline{\xi}_n$ results in an overall more positive gradient, with some of the main differences being the reduction of the sharp negative gradient near the top boundary layer and the inclusion of a positive gradient in the bulk of the flow (in contrast to the almost zero derivative in the original formulation). Therefore, the change in negativity of the term must not be driven by the changes in value of $d\phi/d\zeta$ itself, but by *where* these changes occur.

This is not the behaviour we would expect if the term being explored was the Reynolds stress $\overline{u'w'}$, as it has been shown to be negative across the whole domain in Figure 5. If we were using this term, then the increase in the positivity of the derivative of ϕ would have necessarily meant an increase in negativity of the overall term. However, the Fourier Reynolds stress does not seem to behave like that, causing the drop in negativity.

With all this information, we can confidently say that the correction factor $\overline{\xi}_n$ does not cause the spectral constraint to become negative, therefore being an acceptable solution to the problem. However, acceptable does not mean good.

The original idea of $\overline{\xi}_n$ was to help λ_0 to behave more like the DNS-obtained $\overline{\lambda}$. Figure 19 shows the obtained corrected λ_0' compared to λ_0, $\overline{\lambda}$ and λ_{min}. At low wavenumber, λ_0' is still considerably lower than the DNS results. However, at high wavenumbers it approximates their behaviour much better, albeit with some overshoot.

The behaviour at low wavenumbers is to be expected though: the way QUINOPT works is by seeking the optimisation of the functional $\mathcal{B}$ subject to the positivity of the spectral constraint. If λ_0' behaved exactly like $\overline{\lambda}$, it means that the functional could be further optimised, as $\mathcal{Q}_n$ is still very positive. Therefore, the optimal solution will always be very close to breaking the spectral constraint, but without actually doing so.

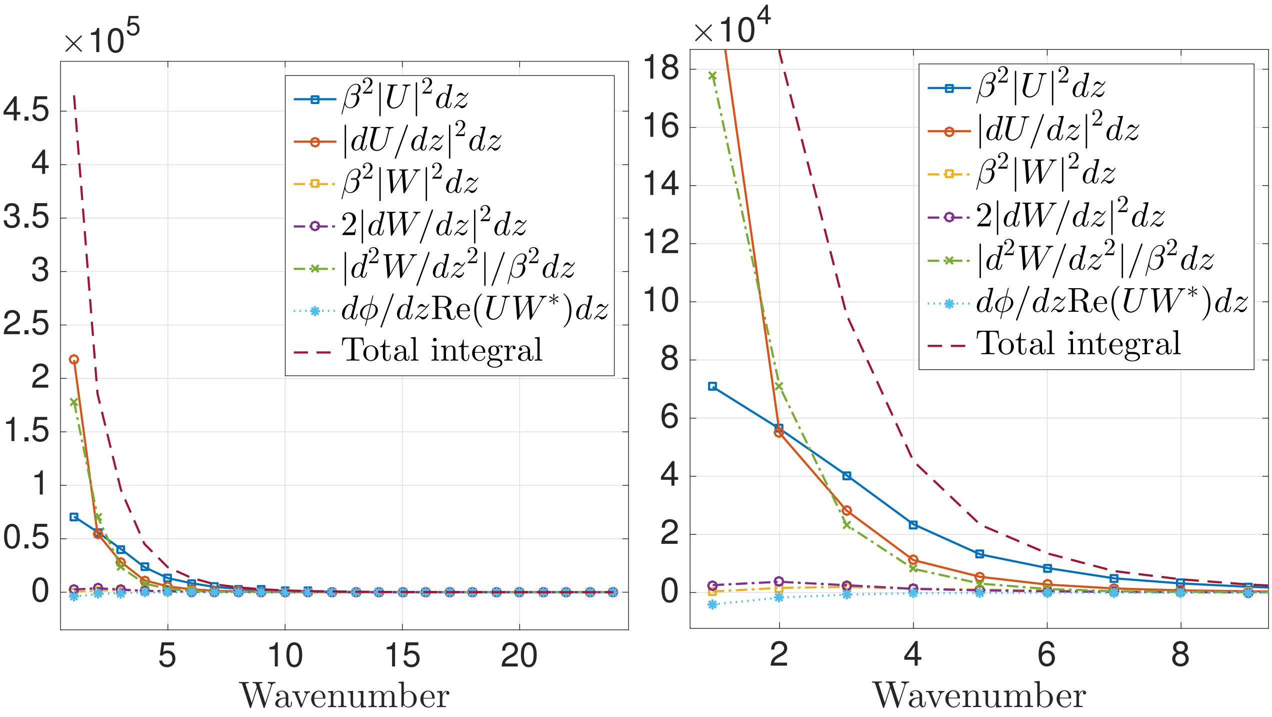

Figure 18: Evolution of each term of $\mathcal{Q}_n$ from DNS with corrected $d\phi_{\overline{\xi}}/d\zeta$, for $Gr = 3844$. Notice that the values are not normalised. All quantities are integrated in the DNS domain $[-0.5, 0.5]$. The plot on the right emphasises the behaviour at low wavenumbers.

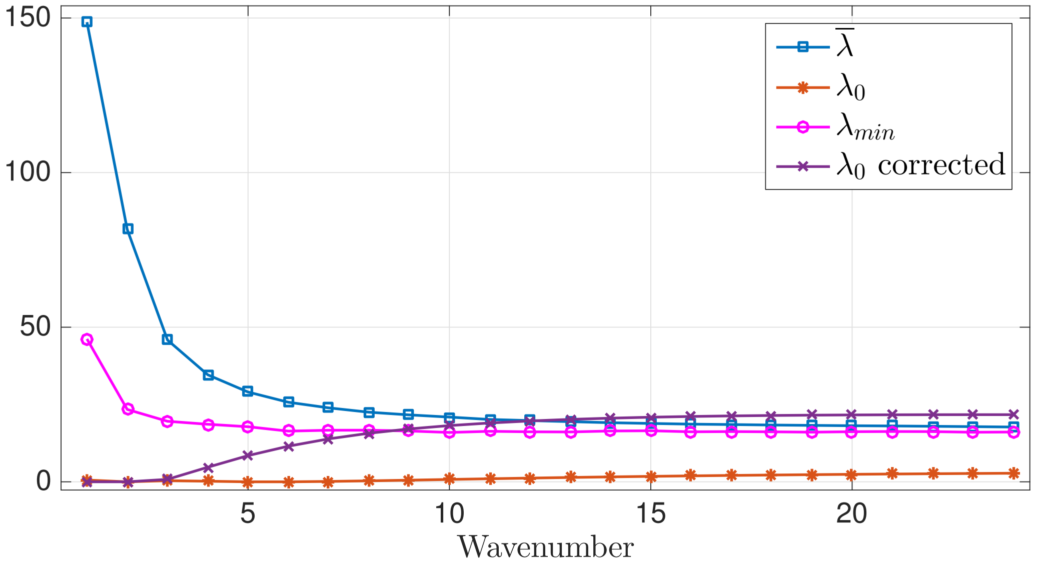

Figure 19: Evolution with wavenumber of λ_0 from QUINOPT, λ_{min} and $\overline{\lambda}$ from DNS, and λ_0 with the correction factor $\overline{\xi}$ applied, for $Gr = 3844$.

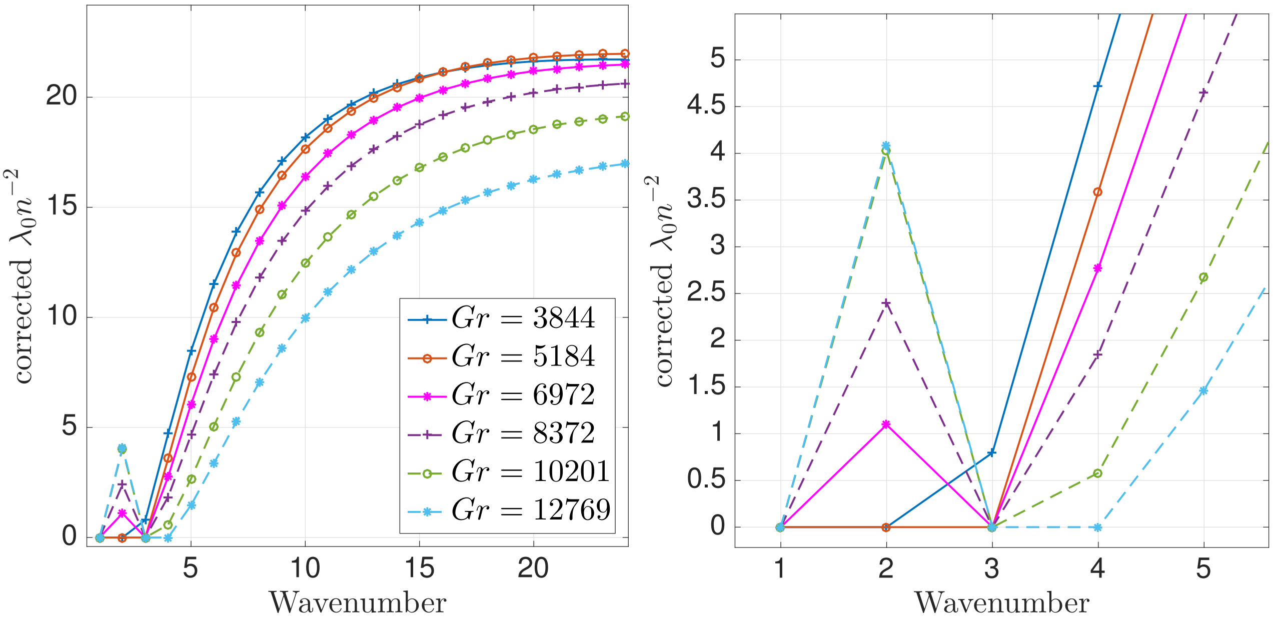

Figure 20: Evolution of λ_0 with wavenumber for a range of Gr. The plot to the right emphasises the behaviour at low wavenumbers, as well as sharing legend with the plot to the left.

This is what causes the unexpected behaviour of λ_0' at low wavenumbers. Remarkably, λ_0' seems to increase monotonically until it reaches what seems to be an asymptotic value at high enough wavenumber. Therefore, applying the correction factor $\bar{\xi}_n$ seems to greatly reduce the impact of the critical wavenumbers on λ_0', as it presents only one bifurcation. Interestingly, the negative term in Figure 18 seems to quickly tend to zero, and its value is smaller compared to the non-corrected background field. This is especially noticeable at $n = 2$, a critical wavenumber, reinforcing the idea that the correction factor is affecting the critical wavenumbers of the system.

Considering that the bifurcations are associated with the internal structure of the background field, this explains why the corrected background field behaves more as the mean flow: $\bar{\xi}_n$ is effectively limiting the bifurcations in the system and therefore the creation of some of the non-physical features of the background field. This can be seen for a range of Gr in Figure 20. When compared to λ_0, the corrected λ_0 is more positive at lower wavenumbers, significantly more positive at higher ones and also presents less bifurcations at high Gr.

IV. Adjusted bounds for C_ε

All this work allows us to further explore potential improvements in the bounds for C_ε. As discussed in the introduction, the latest methods presented by Wynn et al. are considered optimal, meaning there is not further improvement within the variational principle. However, their bounds still lie one order of magnitude above the actual experimental data.

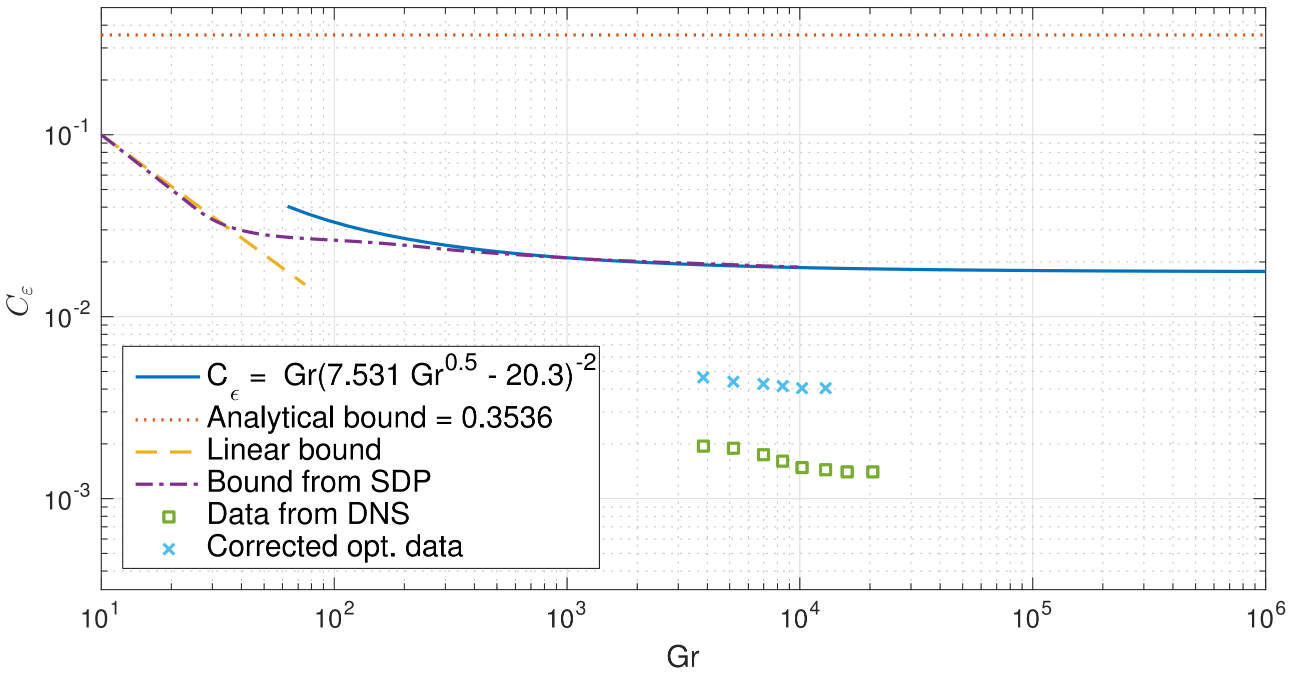

Figure 21: Plot of bounds obtained using different methods, as well as the values for C_ε obtained via DNS

The fact that this variational principle seems exhausted is what motivated us to obtain the correction factor $\overline{\xi}_n$ after studying in detail the behaviour of the spectral constraint, and now we explore whether this correction factor actually improves the upper bounds on C_ε.

We present the data for different bounds in Figure 21. The top bound is the analytical bound obtained by Doering et al. [7]. It can be seen to lie approximately one order of magnitude from the other bounds. The numerical bound $C_\varepsilon = Gr(7.531Gr^{0.5} - 20.3)^{-2}$ was obtained in [20] by Tang et al. by solving computationally a similar problem, but with a body force being applied at the top layer, instead of solving a proper stress driven shear flow.

However, the bound obtained by Wynn et al. is for stress driven shear flows, within the variational principle presented by Doering, and matches the computational bound at least when $Gr > 10^3$. This leads us to believe that stress driven shear flows behave similarly to those obtained by applying a body force at the top layer.

The DNS data is yet another order of magnitude below these bounds. The data plotted corresponds to DNS simulations for $Re_\tau = 62, 72, 84, 92, 101, 113, 133$ and 143.

After obtaining the data from QUINOPT with the correction factor applied it can be said that $\overline{\xi}_n$ does indeed improve the bound, bringing it to the same order of magnitude as the DNS data. This is a significant improvement compared to previously obtained bounds.

However, the corrected data is approximately 2 times larger than the DNS data. This can be attributed to the issues explained during the validation of the correction factor: even though the high wavenumber behaviour of the corrected λ_0 is very similar to the DNS data, the low wavenumber behaviour is still vastly different due to the way the optimisation problem is set up.

We do not believe there are possible improvements within the provided framework in this dissertation. The idea of forcing λ_0 to behave like $\overline{\lambda}$ has provided improvements when compared to the previously obtained bounds, but it has been exhausted.

However, as discussed before, one of the consequences of adding $\overline{\xi}_n$ to $\mathcal{Q}_n$ was that the background field behaved more similar to the mean horizontal velocity $\overline{u}$ obtained from DNS. In fact, when looking at Figure 15, it seems like when $\overline{\xi}_n$ is doubled, ϕ resembles $\overline{u}$ even more.

This seems to imply that the more the background field behaves like the real mean flow, the better the bound will be. This could provide motivation for a redefined correction factor $\xi_{\overline{u}}$ that aimed at making ϕ more "real". However, apart from making the bound match the DNS data, there is no mathematical justification or motivation for this method, and therefore it was not explored.

V. Conclusion

Despite the great interest that mathematically bounding turbulent parameters, there has not been extraordinary progress in the field over the last 25 years, mainly due to the difficulty of the problem.

In this dissertation, the fundamentals of the latest available methods for providing bounds have been explored, with great emphasis on the Doering-Constantin variational principle and the implementation using semidefinite programming of Wynn et al. as they provided the most promising framework to obtain bounds for the dissipation coefficient.

Even though these methods provide the best available mathematically rigorous results, they still do not match experimental data. We firstly provided accurate, validated DNS data for stress driven shear flows, and then proposed a method to try to breach the gap between the experimental data and the bounds available. By analysing different parameters in the variational principle and comparing them to DNS data, we obtained an adjusting factor for the spectral constraint $\mathcal{Q}_n$ so that the minimisers of the spectral constraint behaved more similarly to the DNS $\mathcal{Q}_n$.

With all this, we were able to provide bounds for the energy dissipation coefficient C_ε that are of the same order of magnitude as the experimental data, resulting in a meaningful improvement, when compared to the previously obtained bounds, of approximately a factor of five.

However, there were various limitations. The newly obtained bounds are still twice as large as the experimental values of C_ε. Also, the correction factor $\overline{\xi}_n$ requires of DNS or experimental data, and therefore makes the bound semi-empirical, meaning the mathematical rigorousness of previous implementations is lost.

Finally, the DNS provided a lower limit on the range of Gr, as it was not able to sustain turbulence below $Re_\tau \approx 60$, or $Gr \approx 3600$. Also, QUINOPT provided an upper limit on Gr, as the problem became ill-conditioned due to large, ill-conditioned matrices when Gr was increased over $\approx 5 \times 10^4$ for the three dimensional case. This explains the relatively small range of data provided, which did not allow for clear assertions to be made about the asymptotic regime.

As concluding remarks, we believe the current variational principle to be exhausted, and that there will be no further improvements on C_ε until a new variational principle is proposed which restricts ϕ to more realistic flows.

References

[1] For example, regarding active cooling of turbine blades, look at R. Amano, *Advances in gas turbine blade cooling technology*, WIT Transactions on Engineering Sciences, vol 61, pp 149.

[2] G. Coleman, R. Sandberg, *A Primer on Direct Numerical Simulation of Turbulence: Methods, Procedures and Guidelines*, Aerodynamics & Flight Mechanics Research Group, School of Engineering Sciences, University of Southampton, 2010.

[3] S. Orszag, *Analytical Theories of Turbulence*, Journal of Fluid Mechanics, vol 41, 363.

[4] R. Nicodemus, S. Grossmann, M. Holthaous, Physics *Variational bound on energy dissipation in plane Couette flow*, Physical Review E, vol. 56, 6773.

[5] L. Howard, *Bounds on Flow Quantities*, Annual Review of Fluid Mechanics, vol. 4, 473

[6] P. Constantin, C. R. Doering, *Variational bounds on energy dissipation in incompressible flows: Shear flow*, Physical Review E, vol 49, 4087 185.Phys. Rev. E 49, 4087

[7] G. Hagstrom, C. R. Doering, *Bounds on Surface Stress Driven Shear Flow*, Journal of Nonlinear Science, vol 44, 185.

[8] P. Constantin, C. R. Doering, *Variational bounds in dissipative systems*, Physica D, vol. 82, 221.

[9] B. Wen, G. P. Chini, N. Dianati, C. R. Hagstrom, *Computational approaches to aspect-ratio-dependent upper bounds and heat flux in porous medium convection*, Physics Letters A, vol. 377, 2931.

[10] G. Hagstrom, C. R. Doering, *Bounds on heat transport in Bénard-Marangoni convection*, Physics Letters E, vol. 81, 047301.

[11] C. Doering, P. Constantin, *Energy Dissipation in Shear Driven Turbulence*, Physical Review Letters, vol. 69 1648.

[12] L. Abbot, *Introduction to the Background Field Method*, Acta Physica Polonica, Vol. B13, 1.

[13] C. Nobili, F. Otto, *Limitations of the background field method applied to Rayleigh-Bénard convection*, arXiv:1605.08135.

[14] A. Wynn, G. Fantuzzi, *Optimal Bounds with semidefinite programming: an application to stress driven shear flows*, Physics Letters E, vol. 93, 043308.

[15] L. Vandenberghe, S. Boyd, *Semidefinite Programming*, SIAM Review, Vol. 38, 49.

[16] S. Boyd, L. Vandernberghe, *Semidefinite Programming*, SIAM Review, vol. 38, 49.

[17] M. Goemans, D. Williamson, *Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming*, Journal of the Association for Computing Machinery, Vol. 42, 1115.

[18] Encyclopedia of Mathematics, *Gronwall lemma*, http://www.encyclopediaofmath.org/index.php?title=Gronwall_lemma&oldid=30829.

[19] J.C. Vassilicos, *Dissipation in Turbulent Flows*, Annual Reviews Fluid Mechanics, vol. 47, 95.

[20] W. Tang, C. Caulfield, W. Young, *Bounds on dissipation in stress-driven flow*, Journal of Fluid Mechanics, vol. 510, 333.

[21] G. Fantuzzi, A. Wynn, P. Goulart, A. Papachristodoulou *Optimization with affine homogeneous quadratic integral inequality constraints*, arXiv:1607.04210.

[22] *SeDuMi, optimisation over symmetric cones*, http://sedumi.ie.lehigh.edu, Department of Industrial and Systems Engineering, Lehigh University, last checked in March 2017.

[23] S. Boyd, L. Vandernberghe, *Convex Optimisation*, Cambridge University Press. ISBN 978-0-521-83378-3, 2011.

[24] T. Bewley, *Numerical Renaissance: Simulation, Optimization and Control*, http://numerical-renaissance.com, last checked in March 2017.

[25] C. Canuto, M. Y. Hussaini, A. Quarteroni, T.A. Zang, *Spectral Methods in Fluid Dynamics*, Springer-Verlag, 1987.

[26] S. Djeddi, A. Masoudi, P. Ghadimi, *Numerical Simulation of Flow around Diamond-Shaped Obstacles at Low to Moderate Reynolds Numbers*, American Journal of Applied Mathematics and Statistics, vol. 1, 11.

[27] Andrew Olson's GitHub repository for DNS data, *https://github.com/olsongallardo/Bounds_on_Energy_Dissipation_DNS_data.git*

A. Velocity profiles for a range of Re_τ

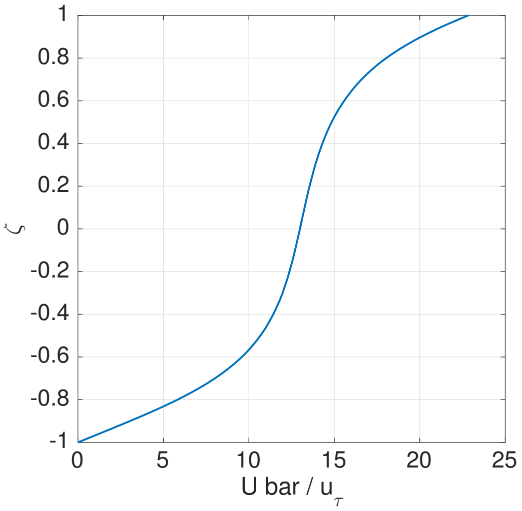

Figure A.1: Plot of the average velocity profile along the vertical domain, normalised by u_τ, for $u_\tau = 4.00 \times 10^{-2}$ and $Re_\tau = 61.49$.

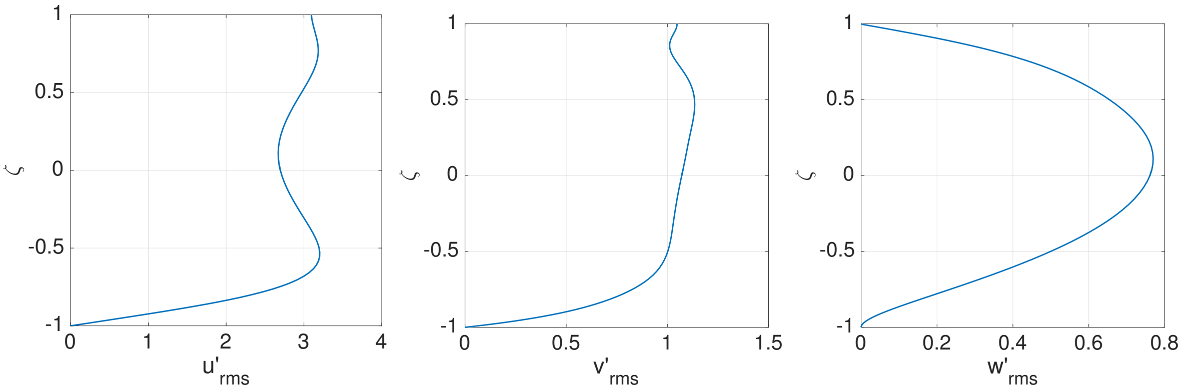

Figure A.2: Plot of the root mean square velocity profiles along the vertical domain, for $u_\tau = 4.00 \times 10^{-2}$ and $Re_\tau = 61.49$. All velocities are normalised by u_τ.

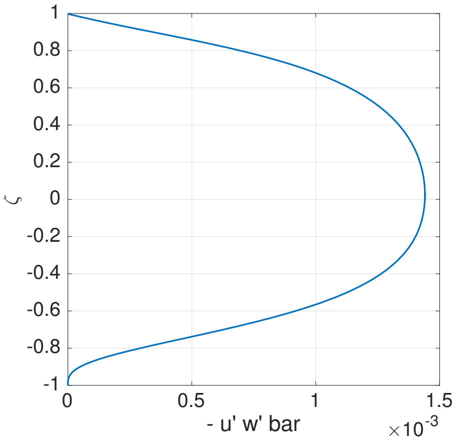

Figure A.3: Plot of the time averaged Reynold stress $\overline{u'w'}$ along the vertical domain, for $u_\tau = 4.00 \times 10^{-2}$ and $Re_\tau = 61.49$.

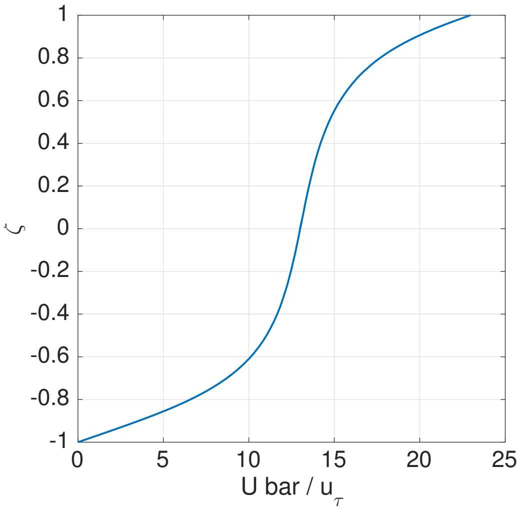

Figure A.4: Plot of the average velocity profile along the vertical domain, normalised by u_τ, for $u_\tau = 3.97 \times 10^{-2}$ and $Re_\tau = 72.14$.

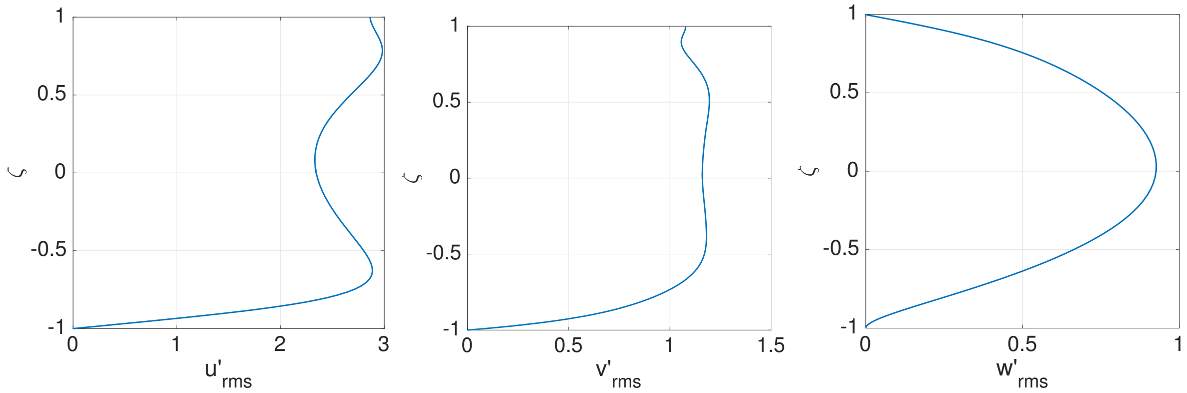

Figure A.5: Plot of the root mean square velocity profiles along the vertical domain, for for $u_\tau = 3.97 \times 10^{-2}$ and $Re_\tau = 72.14$. All velocities are normalised by u_τ.

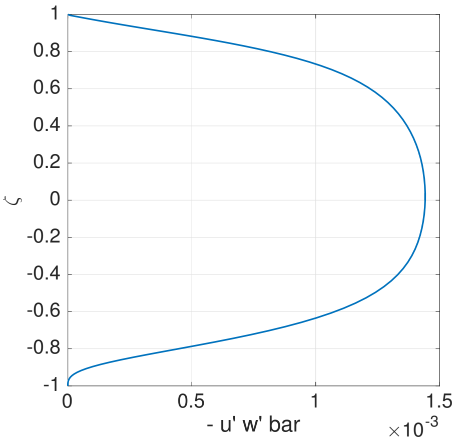

Figure A.6: Plot of the time averaged Reynold stress $\overline{u'w'}$ along the vertical domain, for $u_\tau = 3.97 \times 10^{-2}$ and $Re_\tau = 72.14$.

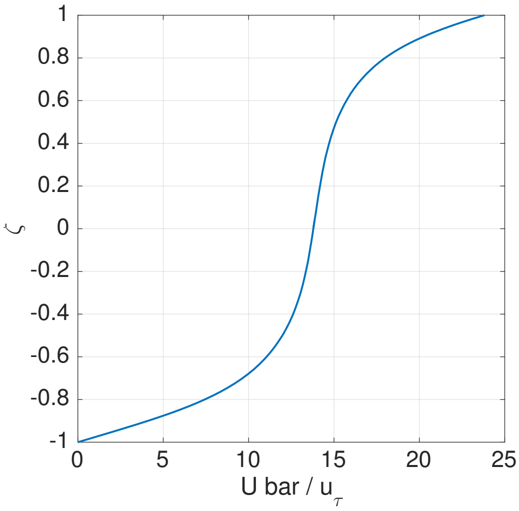

Figure A.7: Plot of the average velocity profile along the vertical domain, normalised by u_τ, for $u_\tau = 3.77 \times 10^{-2}$ and $Re_\tau = 83.70$.

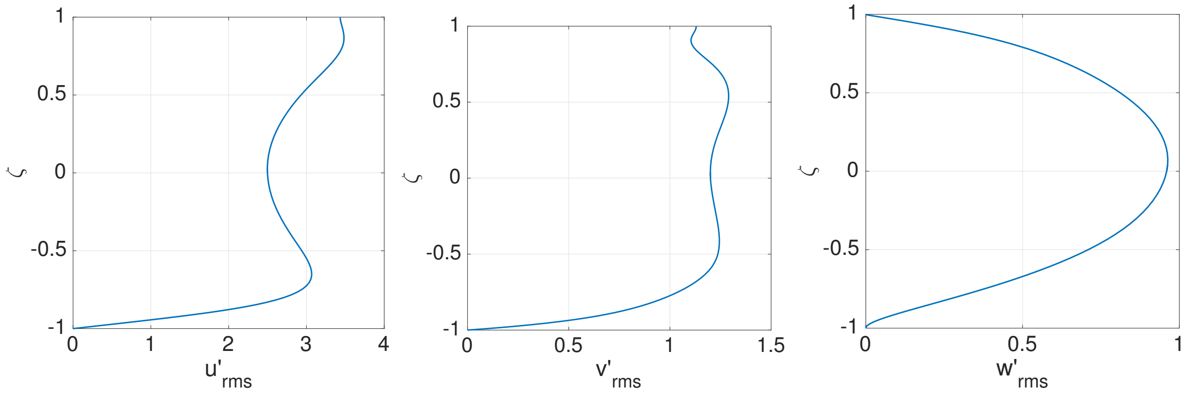

Figure A.8: Plot of the root mean square velocity profiles along the vertical domain, for $u_\tau = 3.77 \times 10^{-2}$ and $Re_\tau = 83.70$.. All velocities are normalised by u_τ.

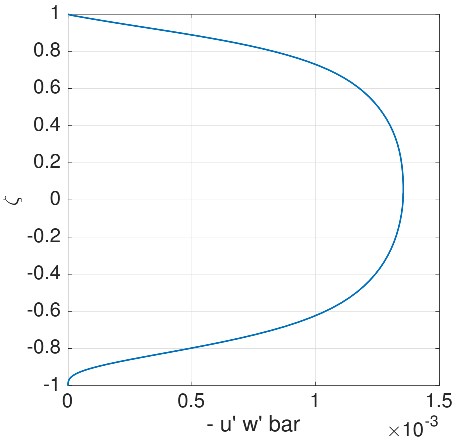

Figure A.9: Plot of the time averaged Reynold stress $\overline{u'w'}$ along the vertical domain, for $u_\tau = 3.77 \times 10^{-2}$ and $Re_\tau = 83.70$..

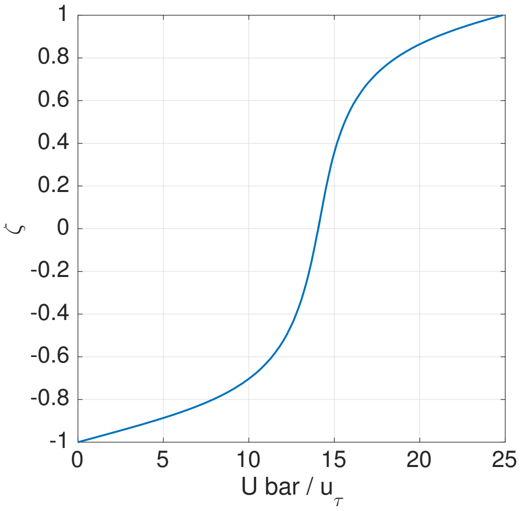

Figure A.10: Plot of the average velocity profile along the vertical domain, normalised by u_τ, for $u_\tau = 3.66 \times 10^{-2}$ and $Re_\tau = 91.52$.

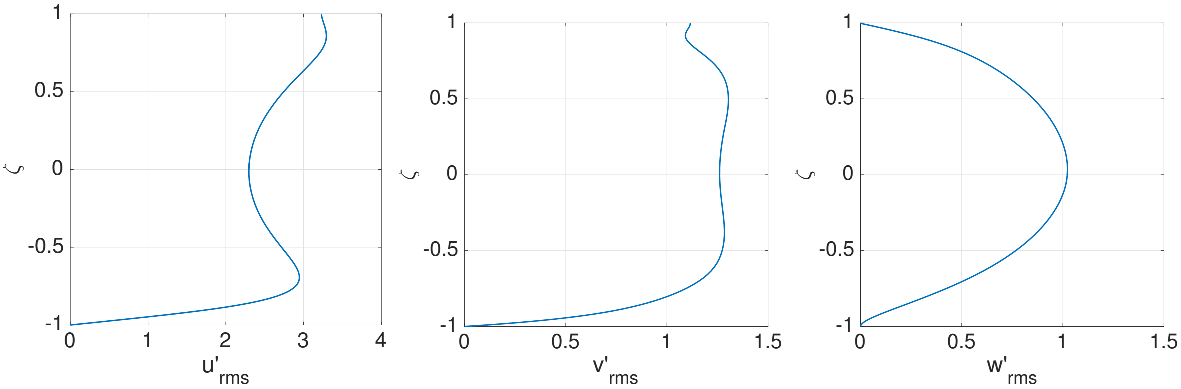

Figure A.11: Plot of the root mean square velocity profiles along the vertical domain, for $u_\tau = 3.66 \times 10^{-2}$ and $Re_\tau = 91.52$. All velocities are normalised by u_τ.

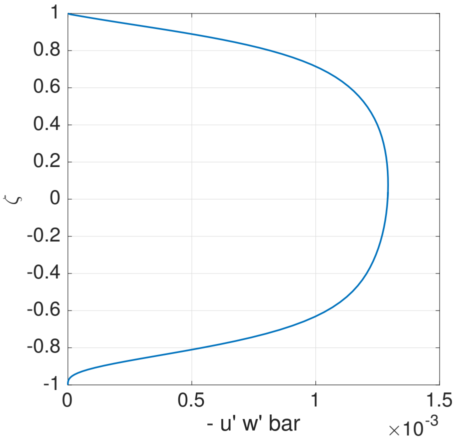

Figure A.12: Plot of the time averaged Reynold stress $\overline{u'w'}$ along the vertical domain, for $u_\tau = 3.66 \times 10^{-2}$ and $Re_\tau = 91.52$.

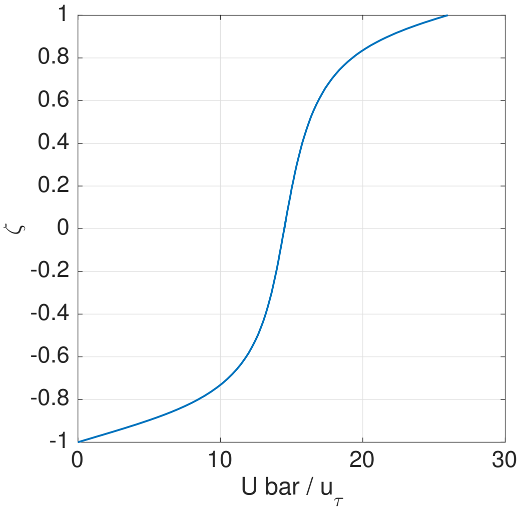

Figure A.13: Plot of the average velocity profile along the vertical domain, normalised by u_τ, for $u_\tau = 3.52 \times 10^{-2}$ and $Re_\tau = 100.77$.

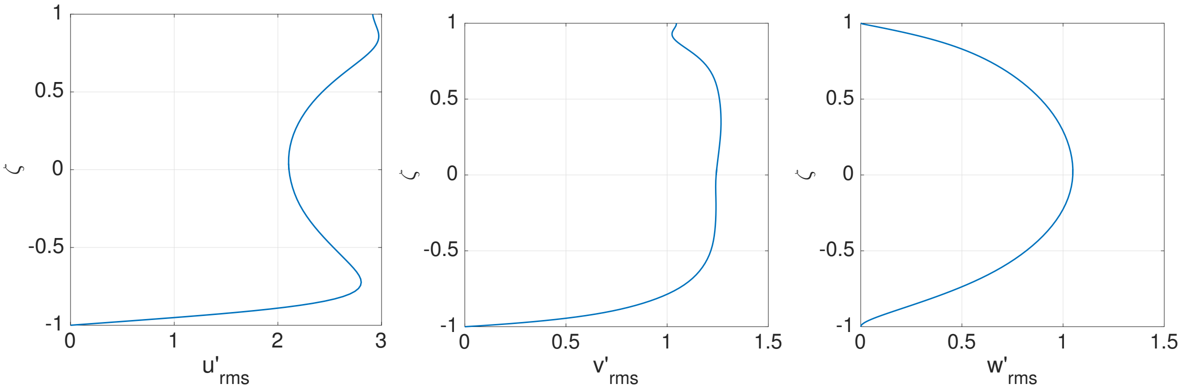

Figure A.14: Plot of the root mean square velocity profiles along the vertical domain, for $u_\tau = 3.52 \times 10^{-2}$ and $Re_\tau = 100.77$. All velocities are normalised by u_τ.

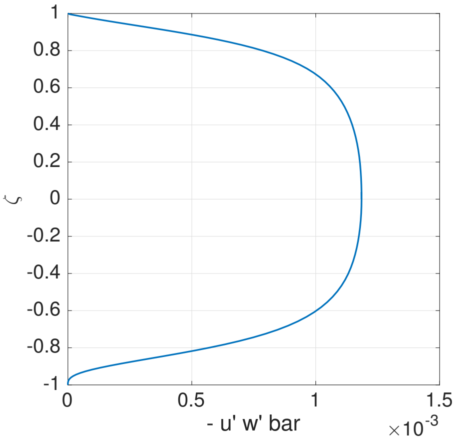

Figure A.15: Plot of the time averaged Reynold stress $\overline{u'w'}$ along the vertical domain, for $u_\tau = 3.52 \times 10^{-2}$ and $Re_\tau = 100.77$.

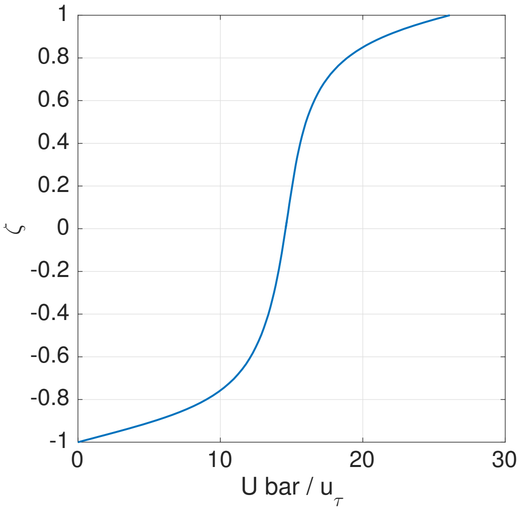

Figure A.16: Plot of the average velocity profile along the vertical domain, normalised by u_τ, for $u_\tau = 3.51 \times 10^{-2}$ and $Re_\tau = 113.26$.

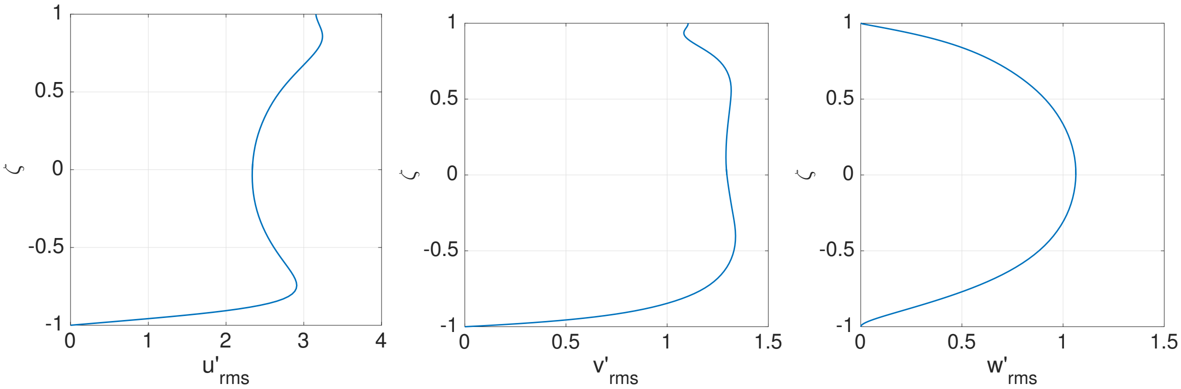

Figure A.17: Plot of the root mean square velocity profiles along the vertical domain, for $u_\tau = 3.51 \times 10^{-2}$ and $Re_\tau = 113.26$. All velocities are normalised by u_τ.

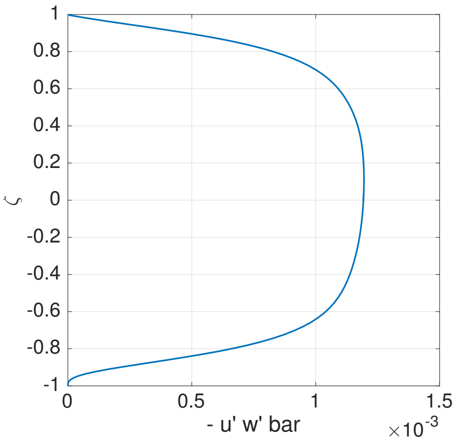

Figure A.18: Plot of the time averaged Reynold stress $\overline{u'w'}$ along the vertical domain, for $u_\tau = 3.51 \times 10^{-2}$ and $Re_\tau = 113.26$.

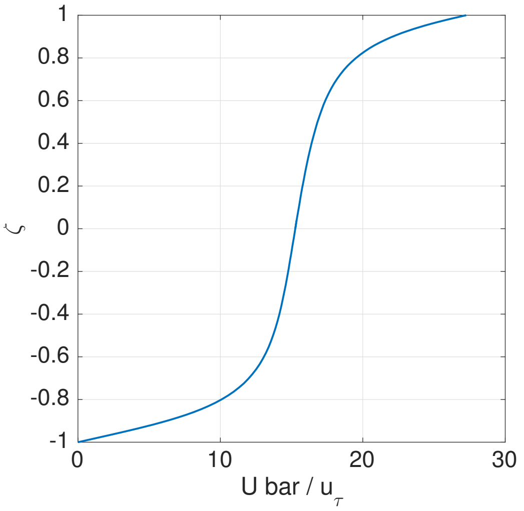

Figure A.19: Plot of the average velocity profile along the vertical domain, normalised by u_τ, for $u_\tau = 3.33 \times 10^{-2}$ and $Re_\tau = 133.29$.

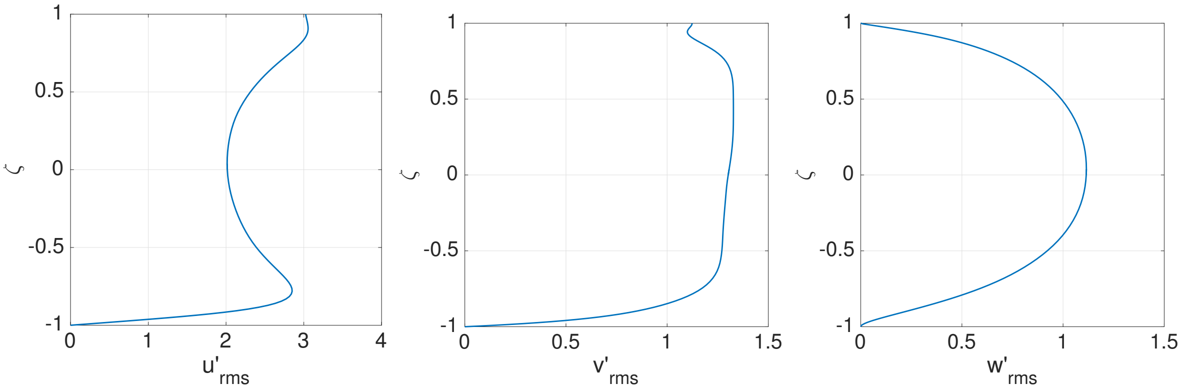

Figure A.20: Plot of the root mean square velocity profiles along the vertical domain, for $u_\tau = 3.33 \times 10^{-2}$ and $Re_\tau = 133.29$. All velocities are normalised by u_τ.

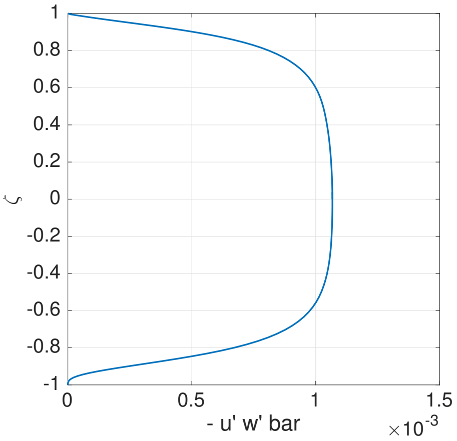

Figure A.21: Plot of the time averaged Reynold stress $\overline{u'w'}$ along the vertical domain, for $u_\tau = 3.33 \times 10^{-2}$ and $Re_\tau = 133.29$.

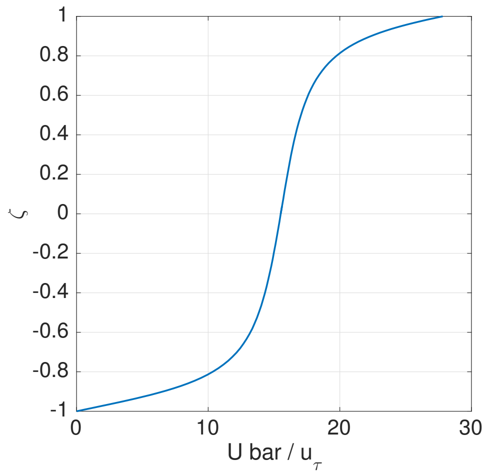

Figure A.22: Plot of the average velocity profile along the vertical domain, normalised by u_τ, for $u_\tau = 3.28 \times 10^{-2}$ and $Re_\tau = 142.70$.

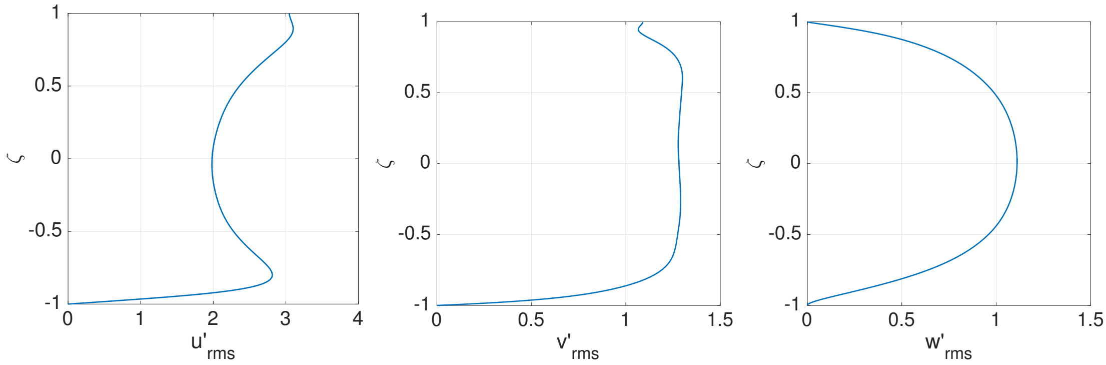

Figure A.23: Plot of the root mean square velocity profiles along the vertical domain, for $u_\tau = 3.28 \times 10^{-2}$ and $Re_\tau = 142.70$.. All velocities are normalised by u_τ.

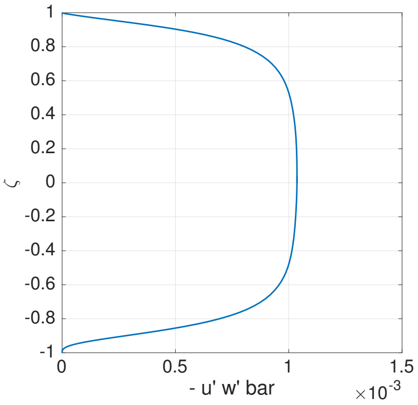

Figure A.24: Plot of the time averaged Reynold stress $\overline{u'w'}$ along the vertical domain, for $u_\tau = 3.28 \times 10^{-2}$ and $Re_\tau = 142.70$..

All these figures are available online as PDFs, as well as the data output from Diablo. To access, click here or, if reading the print version, see Ref. [27].

B. Source FORTRAN code

Diablo is unfortunately not completely open source. Therefore, the only parts of the code included are the parts that were written for this dissertation. Three dots . . . will be used to imply that there is code there, but can't be shown.

The only modifications to the code were made to the subroutine TRACE, which gets called to save statistics based on user input (in our case, every ten timesteps). The code used is the following:

```
      ...

      logical file_exists
      COMPLEX*16  ::  U(0:NX/2, 0:NZ+1, 0:NY+1),
     &                U_y(0:NX/2, 0:NZ+1, 0:NY+1)
      COMPLEX*16  ::  V(0:NX/2, 0:NZ+1, 0:NY+1),
     &                V_y(0:NX/2, 0:NZ+1, 0:NY+1),
     &                V_yy(0:NX/2, 0:NZ+1, 0:NY+1)

      REAL*8  ::  delta1
      REAL*8  ::  delta2

      real BField(0:Ny-1)

      real*8 beta
      real*16 normU(1:NKz)
      real*16 normV(1:NKz)
      real*16 intU(1:NKz)
      real*16 intU_y(1:NKz)
      real*16 intV(1:NKz)
      real*16 intV_y(1:NKz)
      real *16 intV_yy(1:NKz)
      integer n
      complex*16 temp1
      complex*16 temp2
      complex*16 integral

      complex*16 stgV(0:NX/2, 0:NZ+1, 0:NY+1)
      complex*16 stgV_y(0:NX/2, 0:NZ+1, 0:NY+1)
      complex*16 stgV_yy(0:NX/2, 0:NZ+1, 0:NY+1)

      real*16 Q(1:NKz)
      real*16 NegativityTerm(1:Nkz)
      logical flag_trace

      ...
```

```
 !Modification to the TRACE subroutine to obtain the
 !Fourier coefficients of the velocities, as well as
 !the first and second      derivatives. The overall
 !aim is to calculate a quadratic expression
 !Q_n, for each wavenumber and record it in a file.
 !For info about the quadratic expression, see Optimal
 !bounds with semidefinite programming: an application
 !to stress driven shear flows,
 !by Wynn and Fantuzzi, 2016

 !To obtain  Q, we need u, u_y, v, v_y, v_yy,
 !where the subscript denotes derivatives

 !Note, this section will only run if the file containing the
 !derivatives from QUINOPT. The file must be named dPhidzDNS.dat

 !U and V are directly read from cu1, cu2. We need to obtain the
 !derivatives with respect of the vertical axis, using a second
 !order finite difference scheme

 inquire(file = 'dPhidzDNS.dat', EXIST = file_exists)
 write(*,*) file_exists
   beta = -1!initialise beta to use as flag (by definition it
   !cannot be negative
 if (file_exists .EQV. .TRUE.)  then
   !if so that the calculations are only made if the file exists
 U = CU1 ! No need to multiply U
 V = CU2/sqrt(2.0)
 !Multiplying by 1/sqrt(2) due to the way the Fourier transform is
 !performed. Internal domain is 2pi regardless of domain input.
 !Therefore when reading CU values have to take this into
 !consideration. The coefficients are adjusted by multiplying the
 !values by 1/sqrt(2)

 do i = 0, NKx !Using the 2/3rule, NKx = Nx/3
   do k = 0, NKz !Because of the way the variables are packed, the
                 !range 0 to NKz covers all positive wavenumbers
       U_y(i,k,0) = (U(i,k,1) - U(i,k,0))
&       /(GYF(1)-GYF(0))

       V_y(i,k,0) = (V(i,k,1) - V(i,k,0))
&       /(GY(1)-GY(0)) !First der. with forward
```

```
            delta1 = GY(2) - GY(1)
            delta2 = GY(1) - GY(0) !Update dummy variables

            V_yy(i,k,0) = (delta2*V(i,k,2)-V(i,k,1)*(delta1+delta2)
     &         +delta1*V(i,k,0))/(delta1*delta2**2) !Forward 2nd der.

            do j = 1, Ny-1
                delta2 = GYF(j+1) - GYF(j)
                delta1 = GYF(j) - GYF(j-1) !Update dummy variables

                U_y(i,k,j)=(delta1*U(i,k,j+1)+U(i,k,j)*(delta2-delta1)
     &               -delta2*U(i,k,j-1))/(2*(delta1*delta2))

                delta2 = GY(j+1) - GY(j)
                delta1 = GY(j) - GY(j-1) !Update dummy variables

                V_y(i,k,j)=(delta1*V(i,k,j+1)+V(i,k,j)
     &              * (delta2-delta1)-delta2*V(i,k,j-1))
     &              /(2*(delta1*delta2))!Central difference

               V_yy(i,k,j)=(delta1*V(i,k,j+1)-V(i,k,j)
     &          *(delta1+delta2)
     &         +delta2*V(i,k,j-1))/(delta1*delta2**2) !Central 2nd der.

            end do
            U_y(i,k,Ny) = (U(i,k,Ny) - U(i,k,Ny-1)) /
     &       (GYF(Ny) -GYF(Ny-1)) !Last point for U

                delta2 = GY(Ny+1) - GY(Ny)
                delta1 = GY(Ny) - GY(Ny-1) !Update dummy variables

            V_y(i,k,Ny)=(delta1*V(i,k,Ny+1)+V(i,k,Ny)
     &      * (delta2-delta1)-delta2*V(i,k,Ny-1))
     &      /(2*(delta1*delta2))!Central difference


            V_yy(i,k,Ny)=(delta1*V(i,k,Ny+1)-V(i,k,Ny)
     &      *(delta1+delta2)
     &      +delta2*V(i,k,Ny-1))/(delta1*delta2**2) !Central 2nd der.

            V_y(i,k,Ny+1) = (V(i,k,Ny+1)-V(i,k,Ny))
     &       / (GY(Ny+1)-GY(Ny))!And finally backwards difference
```

```
       V_yy(i,k,Ny+1)=(delta1*V(i,k,Ny+1)-V(i,k,Ny)
&      *(delta1+delta2)+delta2*V(i,k,Ny-1))/(delta1*delta2**2) !Back
   end do
 end do



 !Now read the file for the background field
 !dPhidz should be preformatted in MATLAB for computational
 !efficiency. This requires the data to be appropriately scaled
 !(compare DNS domain to QUINOPT domain) as well as adjusted by
 !Reynolds number (different type of non-dimensional N-S used in
 !DNS and Wynn’s method
 open(unit=7323,file= ’dPhidzDNS.dat’)
 n=0
 do n = 0, Ny-1
   read(7323,501) BField(n)
 end do
 close(7323)
 !Finally integrate all together, using a trapezium rule, checking
 !all  wavenumbers in the z direction. This will provide a 1xTNKz
 !array of values of Q_n that will be included in a file

 !Now calculate Q
 !Create a file to record the values of Qn
 open(7233, file=’qn.txt’,action=’write’ )
 !As well as files to record the norm of the velocities
 open(7234, file=’Unorm.txt’,action=’write’ )
 open(7235, file=’Vnorm.txt’,action=’write’ )

 !Also, files for the time evolution of all terms of the
 !Q_n
 open(7236, file=’NegTerm.txt’, action=’write’)!The negative
                                               !term phi*Re(U*V)
 open(7237, file=’U.txt’, action=’write’)!beta*U
 open(7238, file=’U_y.txt’, action=’write’)!dU/dy
 open(7239, file=’V.txt’, action=’write’)!beta*V
 open(7240, file=’V_y.txt’, action=’write’)!dV_dy
 open(7241, file=’V_yy.txt’, action=’write’)!d^2V/dy^2 /beta

 do n = 1, NKz !Loop in spanwise direction, only positive
               !wavenumbers (not 0)

   beta = 2*pi*(n)/Lz !Update the value of beta
   do m = 0,0 !Take only the first wavenumber in m - streamwise
                                                   !direction
       integral = 0
       normU(n) = 0
       normV(n) = 0
       intU(n) = 0
       intU_y(n) = 0
       intV(n) = 0
       intV_y(n) = 0
       intV_yy(n) = 0
       NegativityTerm(n) = 0

       !Need to preallocate final value so that it works in
       !following loop to cancel out staggered grid
       stgV(m,n,Ny) = (V(m,n,Ny+1) + V(m,n,Ny))/2
       stgV_y(m,n,Ny)=(V_y(m,n,Ny+1)+
&                            V_y(m,n,Ny))/2!Values at midpoint
       stgV_yy(m,n,Ny) = (V_yy(m,n,Ny+1) +
&                            V_yy(m,n,Ny))/2
     do j= 0, Ny-1

       !Due to the staggered grid in V, we need to interpolate
       !to obtain midpoint values that we can use to calculate Q
       !Transform from GY fo GYF points(non-staggered grid)

       stgV(m,n,j) = (V(m,n,j+1) + V(m,n,j)) /2
       stgV_y(m,n,j) = (V_y(m,n,j+1)
&                               + V_y(m,n,j))/2!Values at midpoint
       stgV_yy(m,n,j) = (V_yy(m,n,j+1)
&                               + V_yy(m,n,j))/2


       !Trapezoidal integration of each term of the integral for
       !Q(n)

       temp1 = beta**2 * abs(U(m,n,j))**2
       temp2 = beta**2 * abs(U(m,n,j+1))**2
       intU(n) = intU(n) + (temp2 + temp1)*(GYF(j+1)-GYF(j))/2

       temp1 = abs(U_y(m,n,j))**2
       temp2 = abs(U_y(m,n,j+1))**2
       intU_y(n) = intU_y(n) + (temp2 + temp1)*(GYF(j+1)-GYF(j))/2
```

```
            temp1 = beta**2 * abs(stgV(m,n,j))**2
            temp2 = beta**2 * abs(stgV(m,n,j+1))**2
            intV(n) = intV(n) + (temp2 + temp1)*(GYF(j+1)-GYF(j))/2

            temp1 = 2*abs(stgV_y(m,n,j))**2
            temp2 = 2*abs(stgV_y(m,n,j+1))**2
            intV_y(n) = intV_y(n) + (temp2 + temp1)*(GYF(j+1)-GYF(j))/2

            temp1 = abs(stgV_yy(m,n,j))**2/(beta**2)
            temp2 = abs(stgV_yy(m,n,j+1))**2/(beta**2)
            intV_yy(n) = intV_yy(n)+ (temp2 + temp1)*(GYF(j+1)-GYF(j))/2

            temp1 = 2*BField(j)*real(U(m,n,j)*conjg(stgV(m,n,j)))
            temp2 = 2*BField(j+1)*real(U(m,n,j+1)*conjg(stgV(m,n,j+1)))
            NegativityTerm(n) = (temp2+temp1)*(GYF(j+1)-GYF(j))/2
     &                  + NegativityTerm(n)

            temp1 = conjg(U(m,n,j)) * U(m,n,j)
            temp2 = conjg(U(m,n,j+1)) * U(m,n,j+1)
            normU(n) = (temp2+temp1)*(GYF(j+1)-GYF(j))/2 + normU(n)

            temp1 = conjg(stgV(m,n,j)) *stgV(m,n,j)
            temp2 = conjg(stgV(m,n,j+1)) * stgV(m,n,j+1)
            normV(n) = (temp2+temp1)*(GYF(j+1)-GYF(j))/2 + normV(n)


           end do

            !And calculation of the whole integral
            integral = (intU(n) + intU_y(n) + intV(n) + intV_y(n) +
     &                 intV_yy(n) + NegativityTerm(n))
            !Qn is normalised by the norm of the velocities
            Q(n) = 2*integral/(normU(n) + normV(n))

        end do

      end do

      !Now write all calculated values in respective file

      WRITE(7233, 501) TIME, (Q(n), n=1,NKz)!Implied DO loop to write all
      WRITE(7234, 501) TIME, (2*normU(n)/(nu**2), n=1,NKz)
      WRITE(7235, 501) TIME, (2*normV(n)/(nu**2), n=1,NKz)
      WRITE(7236, 501) TIME, (2*NegativityTerm(n)/(nu**2), n=1,NKz)
      WRITE(7237, 501) TIME, (2*intU(n)/(nu**2), n=1,NKz)
      WRITE(7238, 501) TIME, (2*intU_y(n)/(nu**2), n=1,NKz)
      WRITE(7239, 501) TIME, (2*intV(n)/(nu**2), n=1,NKz)
      WRITE(7240, 501) TIME, (2*intV_y(n)/(nu**2), n=1,NKz)
      WRITE(7241, 501) TIME, (2*intV_yy(n)/(nu**2), n=1,NKz)


 501  format(100(1x,e16.10))

      flag_trace = .TRUE.

      else if (flag_trace .NEQV. .TRUE.) then

        write(*,*) "File␣containing␣background␣field␣not␣found␣or␣not
␣␣␣␣␣&␣appropriately␣named␣(should␣be␣dPhidzDNS.dat).␣No␣quadratic
␣␣␣␣␣&␣expression␣being␣calculated"
        flag_trace = .TRUE.
      end if   !End of if statement checking for file

      return
      end subroutine trace .
```