

# Fourier continuation method for incompressible fluids with boundaries<sup>☆</sup>

Mauro Fontana <sup>a\*</sup>, Oscar P. Bruno <sup>b</sup>, Pablo D. Mininni <sup>a</sup>, Pablo Dmitruk <sup>a</sup>

<sup>a</sup> Universidad de Buenos Aires, Facultad de Ciencias Exactas y Naturales, Departamento de Física, & IFIBA, CONICET, Ciudad Universitaria, Buenos Aires, 1428, Argentina

<sup>b</sup> Computing and Mathematical Sciences, Caltech, Pasadena, CA 91125, USA



## ARTICLE INFO

### Article history:

Received 10 March 2020

Received in revised form 17 June 2020

Accepted 29 June 2020

Available online 2 July 2020

### Keywords:

Navier–Stokes equation

Non-periodic boundary conditions

Spectral methods

Fourier continuation

Explicit time-stepping

Simulations

## ABSTRACT

We present a Fourier Continuation-based parallel pseudospectral method for incompressible fluids in cuboid non-periodic domains. The method produces dispersionless and dissipationless derivatives with fast spectral convergence inside the domain, and with very high order convergence at the boundaries. Incompressibility is imposed by solving a Poisson equation for the pressure. Being Fourier-based, the method allows for fast computation of spectral transforms. It is compatible with uniform grids (although refined or nested meshes can also be implemented), which in turn allows for explicit time integration at sufficiently high Reynolds numbers. Using a new parallel code named SPECTER we illustrate the method with two problems: channel flow, and plane Rayleigh–Bénard convection under the Boussinesq approximation. In both cases the method yields results compatible with previous studies using other high-order numerical methods, with mild requirements on the time step for stability.

© 2020 Elsevier B.V. All rights reserved.

## 1. Introduction

Numerically solving the Navier–Stokes equations is one of the standard tools employed for researching the behavior of turbulent flows. As the study of turbulence often requires quantification of high-order statistical quantities, high-order numerical methods are in many cases preferred [1,2]. Moreover, it is believed that at large enough distances from physical boundaries some properties of turbulent flows are universal, and thus studies sometimes concentrate on the dynamics of the bulk of the fluid. This motivates the study of “isotropic and homogeneous turbulence”, a problem which can be successfully modeled employing periodic boundary conditions. This comes with great computational advantages, as the Navier–Stokes equations can be very efficiently solved in periodic domains utilizing Fourier representations and pseudospectral calculations [3–6]. Moreover, if compressibility effects can also be neglected, enforcing the continuity condition in this case reduces to solving a Poisson equation for the fluid pressure with periodic boundary conditions, which in the wavenumber domain can be easily and efficiently accomplished.

Notwithstanding the major importance of understanding bulk dynamics in turbulent flows, the statistics of turbulence near

boundaries is clearly as important, with implications for industrial, geophysical, environmental, and astrophysical flows [7,8]. Classical problems where actual physical boundaries must be considered include the flow through a pipe or a channel maintained either by an imposed pressure difference [9–13] or a moving wall [14–16], and the natural convection occurring when a box is heated at one end and cooled at the other [17–20]. Also, in the case of conducting fluids, Hartmann flows represent a classical example of wall-bounded magnetohydrodynamic (MHD) turbulence [21,22].

However, when the presence of walls must be accounted for, the classical pseudospectral method using fast Fourier transforms (FFTs) as described in [3] cannot be employed, as the Gibbs phenomenon severely degrades (or even forbids) convergence. Consequently, several other high-order representations for the non-periodic directions have been introduced [23], including fully spectral methods [24–27], Chebyshev pseudospectral methods [5, 6,28], B-splines [29,30], and spectral element methods [31–33]. All these techniques have been successfully employed in several scientific solvers for turbulent flows and for other partial differential equations (PDEs). When dealing with incompressible flows, however, directly solving the Poisson equation for the pressure becomes a very computationally demanding task, and can even be ill-behaved for some methods (see, e.g., discussions in [6, 28]). Common strategies to avoid this problem include recasting the equations to another set of variables that automatically enforce incompressibility such as the normal velocity–normal

<sup>☆</sup> The review of this paper was arranged by David W. Walker.

\* Corresponding author.

E-mail address: [mfontana@df.uba.ar](mailto:mfontana@df.uba.ar) (M. Fontana).

vorticity formulation [9], using preconditioned methods [34], or solving the Poisson equation employing methods based on Green functions and integral equations [35–37].

Another approach for simulating incompressible fluids with boundaries while retaining a Fourier representation of the fields is the usage of penalization techniques, as in virtual or immersed boundary methods [38–41]. In these methods, non-physical extra terms are added to the equations in order to impose the boundary conditions. This has the drawback that near the boundaries the order of the approximation is notably lowered, resulting even in slow algebraic global convergence of the solutions in the entire domain (see, e.g., the discussion in p. 205 of Ref. [42]). More recently, a high-order solver entirely based on Fourier representations was presented in [43,44] for compressible flows with non-periodic boundary conditions. In that work, an efficient and high-order Fourier Continuation with Gram polynomials (FC-Gram) technique [45,46] was employed to circumvent the Gibbs phenomenon, and boundary conditions were enforced simply by strong imposition of the conditions in physical space.

In this paper we present an FC-Gram based method for incompressible flows in cuboid non-periodic domains, that produces dispersionless derivatives with spectral accuracy inside the domain, and with very high order convergence at the boundaries. The usage of Fourier basis allows for the introduction of an effective Poisson solver for the pressure that is numerically well behaved and that satisfies the divergence-free condition of the velocity field, and its boundary condition, with high accuracy. Moreover, being Fourier-based, the method has the advantage of being compatible with uniform grids, which in turn allows for efficient explicit time integration without requirements of small time steps for stability when considering flows at high Reynolds numbers. After presenting the method, we validate a three-dimensional numerical implementation in two paradigmatic problems: channel flow, and plane Rayleigh–Bénard convection under the Boussinesq approximation.

The remaining of the text is organized as follows: In Section 2 we introduce the governing equations and the notational conventions. In Section 3 we give a brief summary of the key ideas and advantages of the FC-Gram technique. In Section 4, the proposed numerical method for the incompressible Navier–Stokes equations is presented, including the method to solve for the pressure, as well as low- and high-order time stepping schemes. A parallelization method that scales well with the numerical method presented is discussed in Section 5. While the method is described in detail for the incompressible Navier–Stokes equations, generalizations to other PDEs with solenoidal vector fields, such as the incompressible MHD equations, or the incompressible Boussinesq equations, are straightforward, as illustrated with examples in the following sections. In Section 6, turbulent plane Poiseuille flow simulations are presented as an example to validate our algorithm, while in Section 7 plane Rayleigh–Bénard convection simulations are discussed. Finally, Section 8 presents our conclusions.

## 2. Governing equations

In the simplest configuration, we want to solve the three-dimensional (3D) incompressible forced Navier–Stokes equations

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = -\nabla p + \nu \nabla^2 \mathbf{v} + \mathbf{f}, \quad (1)$$

$$\nabla \cdot \mathbf{v} = 0, \quad (2)$$

in a  $(0, 0, 0) \times (L_x, L_y, L_z)$  domain, over a uniform grid of  $N_x \times N_y \times N_z$  points (see Fig. 1). In Eq. (1)  $\mathbf{v}$  is the velocity field,  $p$  is the pressure,  $\mathbf{f}$  is a solenoidal forcing field,  $\nu$  is the kinematic viscosity, and the fluid mass density  $\rho$  is assumed to be equal

to 1 (in dimensionless units) for simplicity. We are particularly interested in the moderate and low kinematic viscosity cases, in which turbulent behavior takes place. In these cases the Courant–Friedrichs–Lowy (CFL) constraint is dominated by the advection term in Eq. (1), and explicit time stepping is, when allowed by the spatial discretization, the time integration method of choice.

For the velocity field we assume periodic boundary conditions in the  $x$  and  $y$  directions, and no-slip boundary conditions in the  $z$  direction,

$$\mathbf{v}|_{x=0} = \mathbf{v}|_{x=L_x}, \quad (3)$$

$$\mathbf{v}|_{y=0} = \mathbf{v}|_{y=L_y}, \quad (4)$$

$$\mathbf{v}|_{z=0} = \mathbf{v}|_{z=L_z} = \mathbf{0}. \quad (5)$$

The periodic boundary conditions in  $x$  and  $y$  also imply that all derivatives of the velocity field  $\mathbf{v}$  must be periodic,

$$\partial_x^n \mathbf{v}|_{x=0} = \partial_x^n \mathbf{v}|_{x=L_x} \quad \forall n \in \mathbb{N}, \quad (6)$$

$$\partial_y^n \mathbf{v}|_{y=0} = \partial_y^n \mathbf{v}|_{y=L_y} \quad \forall n \in \mathbb{N}. \quad (7)$$

Before proceeding, note the choice made here of the PDE to be solved, as well as of the boundary conditions, is done for the sake of clarity in the presentation of the method. Other PDEs involving constraints in the fields being solenoidal (as, e.g., the MHD equations for which the magnetic field  $\mathbf{B}$  satisfies the condition  $\nabla \cdot \mathbf{B} = 0$ ), or other boundary conditions for the fields, can be implemented using the method described in Sections 3 and 4. In the same spirit, a single non-periodic direction is employed in this work both for clarity and for validation with the extensively researched physical problems considered in Sections 6 and 7. However, the generalization of our method to two or more non-periodic directions is possible.

By taking the divergence of Eq. (1) and utilizing Eq. (2), one gets a Poisson equation for the pressure

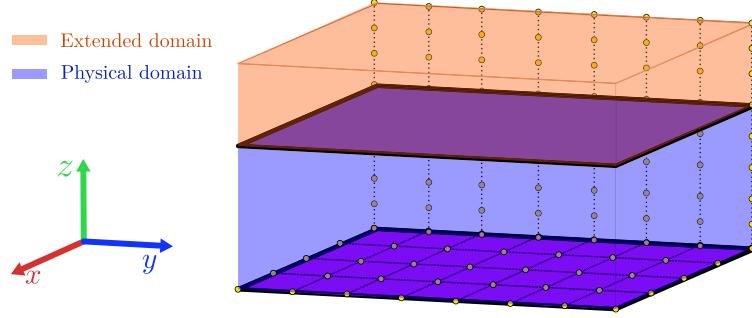
$$\nabla^2 p = -\nabla \cdot [(\mathbf{v} \cdot \nabla) \mathbf{v}]. \quad (8)$$

This implies that the pressure gradient is not an independent variable for incompressible flows, but instead acts as a Lagrange multiplier restricting the velocity field to the subspace of solenoidal fields. However, replacing Eq. (2) with Eq. (8) makes it evident that appropriate boundary conditions compatible with the no-slip condition for  $\mathbf{v}$  must be supplemented to the pressure. The most natural approach is to project Eq. (1) at the boundaries in the direction normal or tangential to the wall, and to solve either a Neumann or a Dirichlet problem. There is no *a priori* reason to assume that both approaches lead to the same solution, a topic which has been of wide discussion and will not be treated here. In the same line of thinking, independent boundary conditions for the pressure could be imposed; we defer the study of this subject for future work. For references on these topics see for example [47–49] and references therein, and the discussion in pp. 83–87 of Ref. [50]. For this work, we project Eq. (1) in the wall normal direction and solve Eq. (8) for the pressure employing Neumann boundary conditions, as discussed in [51].

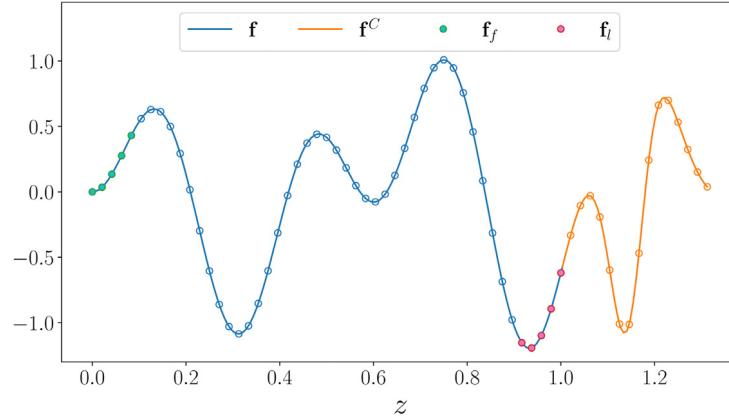
Finally, it is worth mentioning that Eqs. (1), (2), and (5) can be written in several equivalent formulations, such as the normal velocity–normal vorticity formulation employed, for example, in the landmark publication by Kim, Moin and Moser [9]. However, both for simplicity and scalability to other sets of equations (as, for example, Boussinesq or MHD problems), the standard Navier–Stokes formulation in terms of  $\mathbf{v}$  and  $p$  is chosen in this work.

## 3. FC-Gram transformation method

For the spatial discretization in the periodic  $x$  and  $y$  directions we follow the ideas presented in [3] which are now standard



**Fig. 1.** Schematic representation of the domain, with the physical domain in blue, the extended domain used for the FC-Gram method in orange, and the Cartesian axes indicated for convenience. The two horizontal (darker) lids indicate the no-slip boundaries in  $z$ . Boundaries in the  $x$  and  $y$  directions are periodic. A few grid points are shown as a reference. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 2.** The FC-Gram method applied to the function  $f(z) = \sin(15z) \sin(5.5z) \exp(0.3z)$ . The original data  $\mathbf{f}$  (blue line) contains 49 equispaced points in the  $[0, 1]$  interval, and the “continued” data  $\mathbf{f}^C$  (orange line) consists of 15 values (i.e. the number of continuation points is  $C = 15$ ). 5 matching points (i.e.  $d = 5$ ) were used on each boundary, highlighted in green and magenta (respectively indicated as  $\mathbf{f}_f$  and  $\mathbf{f}_l$ ). Note the data in the extended domain is periodic, and thus representable by a Fourier series without Gibbs phenomenon. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

practice in pseudospectral methods for PDEs, in which a trigonometric basis is used for expanding  $v$ ,  $p$  and  $\mathbf{f}$  in the  $(k_x, k_y)$  domain. Using the same basis for the non-periodic  $z$  direction in a straightforward manner would result in the well known Gibbs phenomenon, and produce a major loss of accuracy [52].

The Gibbs ringing problem is circumvented in this work by utilizing a Fourier Continuation technique, in which an extended domain of length  $L'_z$  is considered in the  $z$  direction (see Fig. 1), and an appropriate periodic continuation is computed for all functions in the  $[L_z, L'_z]$  interval using Gram polynomials [43, 45, 46, 53] (see an example in Fig. 2 for a non-periodic function dependent only on  $z$  in the  $[0, 1]$  interval). The union of the original and continued points can then be Fourier transformed to the  $k_z$  space (i.e., expanded into a trigonometric basis). In this way, the error arising from the Gibbs phenomenon is severely reduced. This technique has also the advantage of producing high-order and dispersionless spatial derivatives [54], a property of major importance when dealing with the large range of scales present in highly turbulent flows.

To describe the method in more detail, let us consider a 3D “continued” scalar field  $\phi^c$  (which can be the pressure field  $p$ , each Cartesian component of the velocity field  $v_i$  with  $i = x, y$ , or  $z$ , a temperature field, or the Cartesian components of other vector fields when dealing with other PDEs). The continued scalar field  $\phi^c$  is defined over the  $(0, 0, 0) \times (L_x, L_y, L'_z)$  extended domain, and matches the physical field  $\phi$  on the  $(0, 0, 0) \times (L_x, L_y, L_z)$  region. As fields in the extended domain are 3D-periodic,  $\phi^c$  can

be represented as

$$\phi_{qrs}^c = \sum_{n=m=l=0}^{N_x-1, N_y-1, N_z+C-1} \hat{\phi}_{nml}^c e^{i\mathbf{k}_{nml} \cdot \mathbf{x}_{qrs}}, \quad (9)$$

where  $\hat{\phi}^c$  are the Fourier coefficients of  $\phi^c$ ,  $\mathbf{x}_{qrs}$  is the position vector in the extended domain  $(q\Delta x, r\Delta y, s\Delta z)$  (with  $\Delta x$ ,  $\Delta y$ , and  $\Delta z$  the spatial resolution, and  $q$ ,  $r$ , and  $s$  integers that label each point of the spatial grid),  $\mathbf{k}_{nml}$  is the wavenumber vector

$$\mathbf{k}_{nml} = \frac{\pi(2n - N_x)}{L_x} \hat{\mathbf{x}} + \frac{\pi(2m - N_y)}{L_y} \hat{\mathbf{y}} + \frac{\pi(2l - N_z - C)}{L'_z} \hat{\mathbf{z}}, \quad (10)$$

(assuming an even number of gridpoints), and  $C$  is the number of continuation points in the  $z$  direction.

As mentioned above, to compute the periodic extension of  $\phi(x, y, z)$  in the  $z$  direction we employ the FC-Gram method first presented in [43, 45, 53] and employed with great success in [44, 54–56]. The main idea behind the FC-Gram method is to project the values near the boundaries onto an orthogonal polynomial basis, and to calculate appropriate continuations for each element of the basis. As this is basically a one-dimensional (1D) problem in each direction in which quantities must be extended (i.e., in each direction without periodic boundary conditions), we consider a non-periodic scalar function  $f(z)$  (not to be confused with the forcing  $\mathbf{f}$  defined previously) to explain the method. Let us say  $\mathbf{f}$  is the vector with the values of  $f(z)$  at the grid points in the non-periodic interval  $[0, L_z]$ . Then, in order to generate a periodic

continuation  $\mathbf{f}^c$  from the original  $\mathbf{f}$  values, one needs to specify two parameters, which are the number of continuation points desired  $C$  (i.e., the number of grid points in the extension of the original domain), and the degree of the polynomial basis  $d$  (i.e., the number of values of  $\mathbf{f}$  near each boundary to use in the polynomial adjustment, as illustrated in Fig. 2). Given those parameters, a  $C \times d$  continuation matrix  $A$  and a  $d \times d$  projection matrix  $Q$  can be computed (see definitions of these matrices in [55], and the resulting continuation points are obtained with the expression

$$\mathbf{f}^c = A Q \mathbf{f}_f + A^\dagger Q^\top \mathbf{f}_l, \quad (11)$$

where  $\mathbf{f}_f$  (resp.,  $\mathbf{f}_l$ ) are the first (resp., last)  $d$  points of  $\mathbf{f}$ . In this equation,  $\dagger$  and  $\top$  denote the row-reversing and column-reversing operations, respectively. The resulting data  $\mathbf{f}^c$  and its spatial derivatives are periodic in the  $[0, L_z]$  interval, and thus can be Fourier transformed efficiently. Applying Eq. (11) to each  $s$ -line of  $\phi_{nml}$  provides  $\phi_{nml}^c$ .

It should be noted that both  $A$  and  $Q$  are independent of the original data  $\mathbf{f}$  (or  $\phi$ ), so they can be precomputed only once before starting the iteration of the Navier-Stokes equations and utilized henceforth. Precomputing these tables is extremely fast, demanding only a couple of minutes in a modern CPU core. The computational cost of calculating the continuations is hence reduced to  $2 \times N_x \times N_y$  matrix-vector multiplications with very low dimensionality. Even more, these products can be computed in parallel, and each one fits in the L1 cache of a CPU core. For example, in a case with  $N_z = 991$  we obtain excellent results (both in the order of the approximation as well as in the computational cost) employing  $C = 33$  and  $d = 7$ .

As previously mentioned, the one-dimensional FC-Gram method summarized above can be applied to the 3D case by utilizing Eq. (11) for each line with fixed  $(x, y)$  coordinates and results, for instance, in a velocity field "continued" in the  $z$  direction  $\mathbf{v}^c$ , and whose Fourier coefficients  $\hat{\mathbf{v}}_{nml}^c$  can then be obtained via standard 3D-FFT computations with great accuracy. It is straightforward to see that the FFT and the FC-Gram operators commute and hence  $\hat{\mathbf{v}}^c$  can alternatively be obtained by first transforming  $\mathbf{v}$  to the mixed  $(k_x, k_y, z)$  domain (a 2D-FFT operation) and then performing one dimensional FC-Gram continuations for each  $(k_x, k_y)$  coordinate, followed by an additional 1D-FFT calculation. When computing the 3D-FFT in parallel, this latter property is useful to ensure that the continuations are computed locally, severely boosting performance (see more details in Section 5). Further information on the technical details of the FC-Gram transformation method, and on the mathematical properties of the Gram basis of polynomials, can be found in Appendix B.

#### 4. Method for solving the incompressible Navier-Stokes equations

##### 4.1. Time-splitting forward Euler method and boundary conditions

Numerically, one common practice for dealing with the problem of the boundary conditions for the pressure mentioned in Section 2 is to employ a time-splitting scheme [5,57]. The idea behind this technique is to introduce an auxiliary field  $\mathbf{v}^*$  that verifies the pressureless momentum equation and which, when subtracted the pressure gradient, results in a solenoidal velocity field at the next time step. Although for the simulations in Sections 6 and 7 we employ a higher-order explicit Runge-Kutta integrator, the algorithm for evolving in time the incompressible Navier-Stokes equations will be described here first using a forward Euler method for notational clarity. The generalization to

arbitrary order Runge-Kutta will be deferred until the end of this section.

In the time-splitting forward Euler time stepping method, given the velocity field  $\mathbf{v}^t$  at the time  $t$ , the velocity at the next time step  $\mathbf{v}^{t+\Delta t}$  is obtained as

$$\mathbf{v}^{*t+\Delta t} = \mathbf{v}^t + \Delta t \left[ (\mathbf{v}^t \cdot \nabla) \mathbf{v}^t + \nu \nabla^2 \mathbf{v}^t + \mathbf{f}^t \right], \quad (12)$$

$$\nabla^2 p^{t+\Delta t} = \frac{1}{\Delta t} \nabla \cdot \mathbf{v}^{*t+\Delta t}, \quad (13)$$

$$\mathbf{v}^{t+\Delta t} = \mathbf{v}^{*t+\Delta t} - \Delta t \nabla p^{t+\Delta t}. \quad (14)$$

This method is similar to methods often used to evolve the Navier-Stokes equations in 3D-periodic domains. However, written in this way, boundary conditions can be imposed separately both to  $\mathbf{v}^*$  and  $p$ . This guarantees that after the projection step in Eq. (14), the error of  $\mathbf{v}^{t+\Delta t}$  at the boundary remains controlled for both the normal and tangential directions.

In particular, different boundary conditions can be supplied to both  $\mathbf{v}^*$  after Eq. (12), and to the normal or tangential derivative of  $p$  when solving Eq. (13), depending on the order of approximation desired. For this section we use a modified global  $\mathcal{O}(\Delta t^2)$  scheme derived in [58], which satisfies the boundary conditions with an error of order  $\mathcal{O}(\Delta t)$ . We also employ the stability enhancing modification to the wall-normal velocity presented in [51], where the following boundary conditions were introduced

$$\mathbf{v}_{\parallel}^{*t+\Delta t} = \Delta t \nabla_{\parallel} p^t \quad \text{at } z = 0, L_z, \quad (15)$$

$$\frac{\partial p^{t+\Delta t}}{\partial z} = \frac{1}{\Delta t} \hat{\mathbf{z}} \cdot \mathbf{v}^{*t+\Delta t} \quad \text{at } z = 0, L_z. \quad (16)$$

Here the symbol  $\parallel$  denotes the components tangential to the wall and  $\hat{\mathbf{z}}$  is the unit vector in the  $z$  direction. It should be noted that when coupled with, for example, a second order time integrator (as done in Sections 6 and 7), these boundary conditions result in a  $\mathcal{O}(\Delta t^2)$  slip velocity at the walls, a fact that will be numerically verified.

##### 4.2. Solution for the pressure in the time-splitting forward Euler method

Following the method presented in the previous section, for every timestep we start by solving for the pressureless velocity in Eq. (12). This can be easily accomplished for the continued fields  $\mathbf{v}^{*c}$  in the wavenumber domain, where the evolution equation takes the form

$$\hat{\mathbf{v}}_{nml}^{*c} = \hat{\mathbf{v}}_{nml}^c + \Delta t \left[ \widehat{(\mathbf{v}^c \cdot \nabla) \mathbf{v}^c} \right]_{nml} - \nu k_{nml}^2 \hat{\mathbf{v}}_{nml}^c + \hat{\mathbf{f}}_{nml}^c. \quad (17)$$

As before, hats denote Fourier transformed quantities, and note that time superindices were dropped for clarity. The Fourier coefficients of the non-linear term are obtained by standard pseudospectral calculations in  $\mathcal{O}[N_x N_y N_z \log(N_x) \log(N_y) \log(N_z)]$  operations (i.e., by computing the derivatives in Fourier space, the product in real space, and returning to Fourier space) [3,59]. Computation of derivatives of the continued fields in Fourier space is straightforward, and performed as in pseudospectral methods in periodic domains, by multiplying Fourier coefficients by their corresponding wavenumber and the imaginary unit.

The next step in the algorithm is to transform the velocity field to the  $(k_x, k_y, z)$  domain and apply the boundary conditions in Eq. (15) to the unphysical velocity field  $\mathbf{v}^*$  via strong imposition (also known as "injection"). In other words, values of the pressureless velocity at the boundaries are replaced to fulfill Eq. (15), after which a FC-Gram extension is performed, and  $\mathbf{v}^{*c}$  is transformed back to Fourier space.

Afterwards, the pressure gradient can be obtained by solving Eq. (13) with boundary condition given by Eq. (16). The Poisson

equation accepts a homogeneous solution  $p^H$  satisfying  $\nabla^2 p^H = 0$ , and an inhomogeneous solution  $p^I$  such that  $\nabla^2 p^I$  satisfies Eq. (13). Using periodic boundary conditions in the extended domain, the gradient of the inhomogeneous pressure solution,  $\nabla p^I$ , can be readily computed in the 3D Fourier space as

$$\nabla \hat{p}_{nm}^{lc} = \frac{\mathbf{k}_{nml} \cdot \hat{\mathbf{v}}_{nml}^{*c}}{k^2} \mathbf{k}_{nml}. \quad (18)$$

This solution does not necessarily satisfy the boundary conditions for the pressure, but  $p^H$  can now be used to impose Eq. (16). To compute the homogeneous solution  $p^H$  we take advantage of the fact that in Cartesian coordinates there exists an analytical solution of  $\nabla^2 p^H = 0$  in the mixed  $(k_x, k_y, z)$  domain with coefficients given by

$$\hat{p}_{nm}^H(z) = A_{nm} e^{\gamma_{nm}(z-L_z)} + B_{nm} e^{-\gamma_{nm}z} + Dz. \quad (19)$$

Here,  $\gamma_{nm} = (k_{x,n}^2 + k_{y,m}^2)^{1/2}$ ,  $k_{x,n} = \pi(2n - N_x)/L_x$  and  $k_{y,m} = \pi(2m - N_y)/L_y$  as in Eq. (10), and  $A_{nm}$ ,  $B_{nm}$  and  $D$  are coefficients that depend on the boundary conditions. Differentiating Eq. (19) and using Eq. (16) it is straightforward to get the following expressions for these coefficients

$$A_{nm} = \frac{t_{nm} - b_{nm} e^{-\gamma_{nm} L_z}}{\gamma_{nm} (1 - e^{-2\gamma_{nm} L_z})}, \quad (20)$$

$$B_{nm} = -\frac{b_{nm} - t_{nm} e^{-\gamma_{nm} L_z}}{\gamma_{nm} (1 - e^{-2\gamma_{nm} L_z})}, \quad (21)$$

$$D = \frac{1}{\Delta t} b_{00} = \frac{1}{\Delta t} t_{00}, \quad (22)$$

where

$$b_{nm} = \hat{\mathbf{z}} \cdot \left[ \frac{\hat{\mathbf{v}}_{nm}^*(z)}{\Delta t} - \nabla \hat{p}_{nm}^{lc}(z) \right] \Big|_{z=0}, \quad (23)$$

$$t_{nm} = \hat{\mathbf{z}} \cdot \left[ \frac{\hat{\mathbf{v}}_{nm}^*(z)}{\Delta t} - \nabla \hat{p}_{nm}^{lc}(z) \right] \Big|_{z=L_z}. \quad (24)$$

As before, the notation  $\hat{\mathbf{v}}_{nm}^*(z)$  and  $\nabla \hat{p}_{nm}^{lc}(z)$  denotes quantities are in the mixed  $(k_x, k_y, z)$  domain. Also, note that as the boundary conditions are imposed on the derivative of  $p$ , the total pressure is defined up to a constant that can be safely ignored. An analytical expression for the homogeneous pressure gradient  $\nabla p^H$  in the mixed  $(k_x, k_y, z)$  domain can then be trivially obtained from Eq. (19) as

$$\nabla \hat{p}_{nm}^H(z) = (ik_{x,n} \hat{\mathbf{x}} + ik_{y,m} \hat{\mathbf{y}}) \left[ A_{nm} e^{\gamma_{nm}(z-L_z)} + B_{nm} e^{-\gamma_{nm}z} \right] + \gamma_{nm} \left[ A_{nm} e^{\gamma_{nm}(z-L_z)} - B_{nm} e^{-\gamma_{nm}z} + \frac{D}{\gamma_{nm}} \right] \hat{\mathbf{z}}. \quad (25)$$

We can FC-Gram continue the above expression and Fourier transform in the  $z$  direction to get the coefficients of the “continued” total pressure gradient  $\nabla \hat{p}_{nml}^c = \nabla \hat{p}_{nml}^{lc} + \nabla p_{nml}^{lc}$ , and use Eq. (14) to finally obtain the velocity field at the next time step in the  $(k_x, k_y, k_z)$  domain.

#### 4.3. Filtering

As it is widely known, the computation of non-linear terms by means of pseudospectral calculations generates a pileup of energy in the highest wavenumbers (known as *mode aliasing* in this context), due to the nature of the discrete version of the convolution theorem. For quadratic non-linearities, a standard way of dealing with this instability is by utilizing the so-called 2/3-rule,

in which wavenumbers with  $k > 2k_{\max}/3$  are filtered out before and after the pseudospectral calculation [60], and where  $k_{\max}$  is the Nyquist maximum resolved wavenumber. However, in our context, employing such an abrupt filter would result in a serious distortion of the fields in physical space, as high wavenumbers are required to accurately represent the periodically-extended fields. We circumvent this problem by employing an exponential filter of the kind proposed in [43], such that for a scalar field  $\phi$ , its filtered spectral coefficients  $\hat{\phi}_{nml}^f$  become

$$\hat{\phi}_{nml}^f = \hat{\phi}_{nml} \exp \left[ -\beta \left( \left( \frac{2L_x k_{x,n}}{\pi N_x} \right)^{2p} + \left( \frac{2L_y k_{y,m}}{\pi N_y} \right)^{2p} + \left( \frac{2L_z k_{z,l}}{\pi (N_z + C)} \right)^{2p} \right) \right]. \quad (26)$$

As the largest wavenumber is attenuated by a factor of  $e^{-\beta}$ , it is natural to choose  $\beta = b \ln(10)$ , where  $b$  is the number of significant decimal digits desired (16 for double precision calculations). The parameter  $p$  should be chosen so that the error introduced in the filtering step remains smaller than the one associated to the time-marching scheme. It was shown in [43] that a value of  $2p \geq 55$  suffices to attain an  $\mathcal{O}(\Delta t^5)$  approximation for the case of fourth order Adams–Bashforth integration. For this work we got good results when choosing the value  $2p = 100$ , and this is therefore the value used for all our simulations in Sections 6 and 7.

#### 4.4. Higher-order time-splitting Runge–Kutta method

With the above discussion in mind it is possible now to introduce a full algorithm for a time-splitting  $o$ th order Runge–Kutta integrator, which approximates the no-slip boundary condition with an accuracy  $\mathcal{O}(\Delta t^o)$ . The method we present also has the advantage of requiring low storage, with only a few arrays stored in computer memory in each iteration:

**FOR**  $j$  in  $1, \dots, o$ :

1. Evolve the pressureless momentum equation in  $(k_x, k_y, k_z)$  space

$$\hat{\mathbf{v}}_{nml}^{*t+j\Delta t/o} = \hat{\mathbf{v}}_{nml}^t + \frac{j\Delta t}{o} \left[ \left[ \overbrace{(\mathbf{v}^{t+(j-1)\Delta t/o} \cdot \nabla) \mathbf{v}^{t+(j-1)\Delta t/o}}_{nml} \right] - \nu k_{nml}^2 \hat{\mathbf{v}}_{nml}^{t+(j-1)\Delta t/o} \right], \quad (27)$$

where the continuation superindices  $^c$  were dropped for clarity. Note that the filter in Eq. (26) must be applied to the non-linear term to prevent aliasing instabilities.

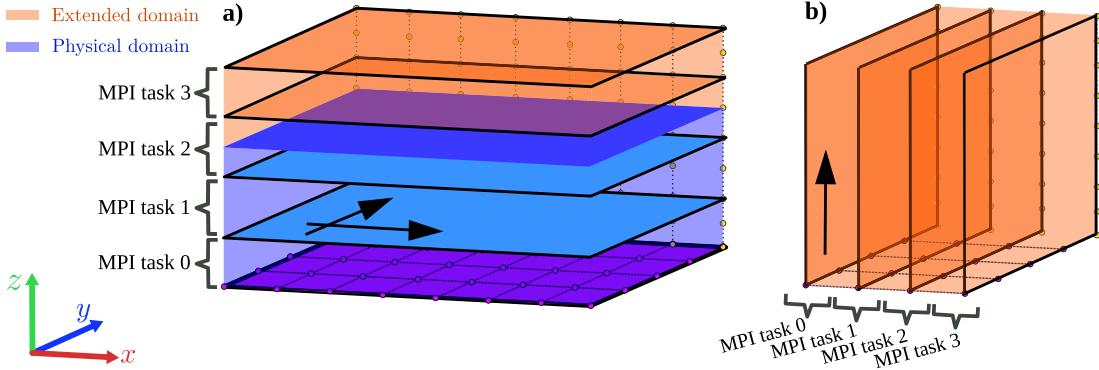
2. Transform  $\hat{\mathbf{v}}^{*t+j\Delta t/o}$  to the  $(k_x, k_y, z)$  domain, and impose

$$\mathbf{v}_{\parallel}^{*t+j\Delta t/o} = \frac{j\Delta t}{o} \nabla_{\parallel} p^{t+(j-1)\Delta t/o} \quad \text{at } z = 0, L_z. \quad (28)$$

3. FC-Gram transform  $\mathbf{v}^{*t+j\Delta t/o}$  back to the  $(k_x, k_y, k_z)$  domain, and obtain the inhomogeneous solution for the pressure

$$\nabla \hat{p}_{nml}^{lt+j\Delta t/o} = \frac{\mathbf{k}_{nml} \cdot \hat{\mathbf{v}}_{nml}^{*t+j\Delta t/o}}{k^2} \mathbf{k}_{nml}. \quad (29)$$

4. Transform  $\hat{\mathbf{v}}^{*t+j\Delta t/o} - \nabla \hat{p}_{nml}^{lt+j\Delta t/o}$  to the  $(k_x, k_y, z)$  domain, and use Eqs. (19) to (21) to solve for the homogeneous



**Fig. 3.** Slab decomposition used for the parallelization. Notation is the same as in Fig. 1. (a) The physical domain, with the extended domain in real space. Each MPI task gets one of the slabs (for convenience, only four tasks are shown). Note FFTs in the  $x$  and  $y$  directions (indicated by the black arrows) can be computed locally with this decomposition. (b) Arrays in Fourier space ( $k_x, k_y, k_z$ ) or in mixed space ( $k_x, k_y, z$ ). After the transposition, with this slab decomposition FFTs in the  $z$  direction are local, and can be computed without communication. Note the full array in  $k_x$  has half the size of the physical domain in  $x$ , as Fourier transformed arrays are complex while physical data is real.

solution for the pressure, now with  $D = ob_{00}/(j\Delta t) = ot_{00}/(j\Delta t)$ , and with the coefficients stemming from the Neumann boundary conditions now taking the form

$$b_{nm} = \hat{\mathbf{z}} \cdot \left[ \frac{o}{j\Delta t} \hat{\mathbf{v}}_{nm}^*(z) - \nabla \hat{p}_{nm}^l(z) \right]_{z=0}, \quad (30)$$

$$t_{nm} = \hat{\mathbf{z}} \cdot \left[ \frac{o}{j\Delta t} \hat{\mathbf{v}}_{nm}^*(z) - \nabla \hat{p}_{nm}^l(z) \right]_{z=L_z}.$$

5. FC-Gram transform the homogeneous pressure gradient to the  $(k_x, k_y, k_z)$  domain, and project the velocity to the space of solenoidal functions

$$\mathbf{v}^{t+j\Delta t/o} = \mathbf{v}^{*t+j\Delta t/o} - \frac{j\Delta t}{o} \nabla p^{t+j\Delta t/o}. \quad (31)$$

end FOR.

## 5. Numerical implementation

The ideas presented in the previous section can be easily implemented in existing parallel pseudospectral codes. For this study we developed a software named Spectral PEriodic Continuation Turbulence SolVER (SPECTER), freely available at <http://github.com/mfontanaar/SPECTER>. SPECTER is a hybrid OpenMP-MPI-CUDA parallel code written in Fortran 95/2003 (with bindings in C and CUDA), built atop the structure of the Geophysical High-Order Suite for Turbulence (GHOST) whose parallelization strategy has been shown to present near optimal scaling for over 200.000 CPU cores and 15.000 GPUs [61] (see also <http://github.com/pmininni/GHOST>).

The algorithm presented in Section 4.4 is easy to implement using a serial FFT. In practice, we use FFTW [62] or cuFFT [63] to compute 1D Fourier transforms. FC-Gram continuation operations are encapsulated in the Fourier transform subroutines that call FFTW or cuFFT. Once a pressure solver and FC-Gram routines are implemented, the rest of the code is remarkably similar to any other pseudospectral code commonly used to study isotropic and homogeneous turbulence. As a result, two advantages of the method follow: (1) Obtaining solutions to the evolution of turbulent flows with boundaries in cuboid domains becomes relatively inexpensive (compared with methods for flows in fully periodic domains), as the overhead of computing FC-Gram continuations and of finding a homogeneous solution for the pressure is relatively small. (2) Efficient parallelization of the method is straightforward, as parallelization methods developed for pseudospectral periodic codes can be easily extended to our method.

In particular, SPECTER uses a 1D domain decomposition (also called a “slab” decomposition) for the parallelization [61]. The domain in real space is partitioned in the  $z$  direction, while the Fourier space is partitioned in the  $k_x$  direction (see Fig. 3). Each MPI task operates locally over one of these slabs, and MPI is used for operations that require cross-task communication. In particular, most of the communication in the code is done when the 3D-FFTs are computed: in the 1D domain decomposition FFTs of real arrays in the  $x$  and  $y$  directions are “local” (i.e., each MPI task has all the information to compute the FFTs in these directions using a serial FFT library), while FFTs in the  $z$  direction are non-local. To solve this problem, parallel pseudospectral codes perform a transposition of the data: data is re-arranged in such a way that arrays in Fourier space, both in the  $(k_x, k_y, k_z)$  and in the mixed  $(k_x, k_y, z)$  spaces, are partitioned in the  $k_x$  direction and thus the  $z$  direction becomes “local” for the FFTs (i.e., each MPI task has all the information required to compute the FFTs in this direction in their portion of the domain independently of the other tasks). The transposition involves an all-to-all communication, and is handled in SPECTER using the same techniques as in GHOST (see [61] for more details).

To increase the number of processors that can be used for the parallelization, operations in each MPI task can be further parallelized using OpenMP. To this end, FFTs in each MPI task are performed using multiple threads with FFTW, and all other loops are parallelized using OpenMP pragmas. Moreover, if instead of FFTW, cuFFT is used, FFTs can be computed in CUDA-enabled graphical processing units (GPUs). As this multi-level hybrid parallelization scheme has been described elsewhere, and requires no modification to the Fourier Continuation method presented here, we refer the reader to [61] for a detailed description as well as for scaling studies. It suffices to say that tests of SPECTER in parallel environments show no degradation in the parallelization efficiency when compared with the results in [61], and that as a result we expect our method to scale well as the number of processors is increased.

## 6. Plane Poiseuille flow simulations

As a validation of the method presented above, we now study the flow between two parallel planes driven by a homogeneous pressure difference along the  $x$  (streamwise) direction. The domain is the same as in Fig. 1. This set up is commonly known as plane Poiseuille flow, or simply, channel flow. It is a traditional problem in wall-bounded turbulence, and has been extensively researched both numerically [9,13,64–67] and experimentally [16,68–70]. From the numerical point of view, as

**Table 1**

Parameters of the channel flow simulations. “Run” labels each simulation,  $N_x \times N_y \times N_z$  gives the linear resolution in each direction, and  $C_z$  and  $d_z$  are the number of continuation and boundary matching points in the  $z$  direction, respectively.  $\overline{\partial_x p}$  is the mean pressure gradient in the  $x$  direction,  $\nu$  the kinematic viscosity,  $Re$  is the Reynolds number,  $Re_c$  is the centerline Reynolds number, and  $Re_\tau$  is the Reynolds number based on the friction velocity  $u_\tau$ .

Run	$N_x \times N_y \times N_z$	$C_z$	$d_z$	$\overline{\partial_x p}$	$\nu$	$Re$	$Re_c$	$Re_\tau$
L1	$32 \times 16 \times 39$	25	5	$2.5 \times 10^{-2}$	$2.5 \times 10^{-3}$	330	260	23
L2	$128 \times 64 \times 231$	25	5	$1 \times 10^{-2}$	$1 \times 10^{-3}$	790	580	35
L3	$256 \times 128 \times 479$	33	7	$5 \times 10^{-3}$	$5 \times 10^{-4}$	1490	1010	52
T1	$256 \times 128 \times 479$	33	7	$2.5 \times 10^{-3}$	$2.5 \times 10^{-4}$	2120	1310	77
T2	$384 \times 192 \times 735$	33	7	$2.5 \times 10^{-3}$	$1.5 \times 10^{-4}$	3480	2050	118
T3	$512 \times 256 \times 991$	33	7	$2.5 \times 10^{-3}$	$1.25 \times 10^{-4}$	4270	2500	146
T4	$768 \times 384 \times 1503$	33	7	$2.5 \times 10^{-3}$	$1 \times 10^{-4}$	5380	3125	173

the flow develops a boundary layer near the walls, it may be argued that a method that refines the grid near the walls (as, e.g., Chebyshev-based pseudospectral methods) can be better suited to study this problem. However, our main aim in this section is to validate the numerical method by comparing with previous results, and thus we consider channel flow as a paradigmatic example. To properly resolve the boundary layer with a uniform grid we will use larger spatial resolutions in the  $z$  direction than in  $x$  and  $y$ ; note also that as the method we present here can deal in principle with any boundary condition, a refined grid near the walls could be implemented, matching the solutions using FC-Gram transforms between the grid in the center region of the channel and the grid in the near-wall regions. Finally, in spite of the limitations in the resolution, the regular grid allows us to use explicit time stepping with a mild CFL condition, thus partially compensating for extra computational costs.

Imposing a pressure difference between two ends of the box results in a non-periodic pressure inside the domain in the  $x$  direction. However, the pressure gradient remains periodic, and can be decomposed as  $\nabla p = \nabla p' + \overline{\partial_x p} \hat{x}$ , where  $\overline{\partial_x p}$  is the mean pressure gradient in the  $x$  direction,  $\overline{\partial_x p} = (p|_{x=L_x} - p|_{x=0})/L_x$ . The effect of the pressure difference at the sides of the box is hence equivalent to a constant forcing in the  $x$  direction, and the resulting equations are

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = -\nabla p' + \nu \nabla^2 \mathbf{v} - \overline{\partial_x p} \hat{x}, \quad (32)$$

$$\nabla \cdot \mathbf{v} = 0. \quad (33)$$

This system admits an analytical laminar solution in the limit in which the non-linearities are negligible, given by

$$\mathbf{v} = -\frac{\overline{\partial_x p}}{2\nu} z(z - L_z) \hat{x}. \quad (34)$$

However, as viscosity decreases and non-linear effects become more relevant, the flow is no longer laminar and turbulence develops. To quantify the strength of non-linear effects, the Reynolds number  $Re$  is commonly employed. This dimensionless parameter is defined as the ratio between non-linear and diffusive terms

$$Re = \frac{L_z \overline{v_x}}{\nu}, \quad (35)$$

where  $\overline{v_x}$  is the vertically averaged streamwise speed. Alternatively, one can define a centerline Reynolds number as

$$Re_c = \frac{\delta v_x|_{z=\delta}}{\nu}, \quad (36)$$

where  $\delta = L_z/2$  is the box half-height. Although the exact value of the critical Reynolds number  $Re_{crit}$  for which the system becomes turbulent depends on the way in which the laminar solution is perturbed (as well as on the precise definition of the Reynolds), typical values for  $Re_{crit}$  are in the range  $\approx 1700$  to  $2300$  [7].

Using the SPECTER code we solved Eqs. (32) and (33) inside a box of size  $L_x \times L_y \times L_z = 4\pi \times 2\pi \times 1$ , with  $z$  the only non-periodic

direction. A total of seven direct numerical simulations were performed, for Reynolds numbers  $Re$  ranging from 300 to 5500, resulting in both laminar and turbulent solutions. A summary of the parameters and the resolution used in each simulation is presented in Table 1. For all the laminar simulations (L1 to L3) the ratio  $\overline{\partial_x p}/\nu$  was maintained equal to 10, in order for the maximum velocity to be of order 1, whereas for the turbulent runs (T1 to T4) the mean pressure gradient  $\overline{\partial_x p}$  was kept constant across runs, leaving the centerline velocity as a free parameter. All simulations were integrated in time using a second order version of the method presented in Section 4.4.

The first laminar simulation L1 was started from a fluid at rest, and was evolved in time until a steady state was reached. We verified the agreement of the velocity profile with the analytical solution in Eq. (34). The last output of this simulation was then scaled up to the resolution of run L2 using spectral interpolation, by zero padding the transformed fields in the wavenumber domain. To the resulting velocity field, a random perturbation with 10% amplitude was added at the largest scales, and used as initial condition for the higher Reynolds simulation L2. Generation of a random perturbation that is solenoidal and satisfies the boundary conditions requires some care; in Appendix A we describe a method that allows generation of noise under these conditions. This procedure was subsequently repeated to generate the initial conditions for run L3 from the steady state of run L2, and for run T1 from the last output of run L3. For all laminar simulations (L1 to L3), a steady parabolic profile was obtained at late times for the streamwise velocity  $v_x$  (not shown). For the case of simulation T1, after evolving the system for more than 100 large-scale turnover times, it was clear that a convergence towards a parabolic profile was no longer observed, and a turbulent-like regime was obtained instead. The transition at  $Re \approx 2000$  is compatible with the experimental and numerical studies mentioned above. From the last output of run T1 we started (after rescaling) run T2, without a perturbation as the flow was not laminar anymore, and the same procedure was used to start the following runs.

For the first turbulent simulation T1 we also performed time evolution of the flow employing four different timesteps, namely  $\Delta t = 1.25 \times 10^{-3}$ ,  $5 \times 10^{-4}$ ,  $2.5 \times 10^{-4}$ , and  $1.25 \times 10^{-4}$ , in order to study the dependence of the error in the slip velocity. As a reference, note the CFL condition requires  $\Delta t \leq 2 \times 10^{-3}$ . The result of this analysis is presented in Fig. 4 where the mean squared streamwise velocity at the wall  $\langle u^2|_{wall} \rangle$  is shown as a function of the timestep, along with the predicted scaling of  $\Delta t^{-4}$  (corresponding to the expected  $\Delta t^{-2}$  scaling for the r.m.s. error). A good level of agreement with the theoretical prediction is found. Performing the same analysis for the spanwise wall velocity leads to the same results. On the other hand, the error in the mean squared normal velocity at the wall  $\langle w^2|_{wall} \rangle$  was found to be  $\mathcal{O}(10^{-35})$  and, as expected, independent of  $\Delta t$ . This behavior was also observed in the mean squared divergence  $\langle (\nabla \cdot \mathbf{v})^2 \rangle$ , for which a timestep-independent value of  $\mathcal{O}(10^{-16})$  was obtained in all these simulations at fixed spatial resolution.

Also for simulation T1, we studied the spatial convergence of the method as the vertical resolution was increased, by comparing results with increasing resolution against a high resolution numerical solution with the same parameters. To this purpose, we performed simulations with a varying number of vertical grid points in the extended domain, namely  $N'_z = 512, 768, 1024$ , and 1536, while fixing the number of horizontal grid points to  $N_x \times N_y = 256 \times 128$ , the matching points to  $d_z = 7$ , and the continuation points to  $C_z = 33$  (note these choices result in  $N_z = N'_z - 33$  vertical grid points in the physical domain). The time step was fixed in all simulations to  $\Delta t = 2 \times 10^{-4}$ . All simulations started from the same initial condition, corresponding to an output of simulation T1 at  $t = 82$ , which was converted to the required grid size in each case employing spectral interpolation. After integration for one large-scale turnover time, the final states of the simulations with  $N'_z = 512, 768$ , and 1024 were interpolated spectrally to the grid with  $N'_z = 1536$ . The spatial average of the  $L_1$  pointwise errors ( $\langle |v_i(\mathbf{x}, t) - v_i^{\text{ref}}(\mathbf{x}, t)| \rangle$ ) (where  $i$  is the velocity field component) was then computed using the simulation with  $N'_z = 1536$  as the reference solution  $v^{\text{ref}}$ . The result is shown in Fig. 5, together with the mean  $L_2$  error in the divergence of the velocity field for all values of  $N'_z$ . High order convergence is observed. Note that we focused here in errors as a function of the vertical resolution. Convergence as a function of horizontal resolution is not shown, as a traditional Fourier pseudospectral method is used in  $x$  and  $y$ , and convergence of this method is well characterized in, e.g., [5]. For details of convergence of the Runge–Kutta method as  $\Delta t$  is varied, see also [5].

### 6.1. Analysis of the turbulent simulations

We now discuss the results for the turbulent simulations T1 to T4. To this purpose it is useful to introduce the turbulent velocity field components  $v'_i = v_i - V_i$ , with  $V_i = \langle v_i \rangle_{x,y,t}$  the  $i$ th velocity component averaged over  $x$ ,  $y$  and  $t$ , as well as the standard notation  $u = v_x$ ,  $v = v_y$ ,  $w = v_z$  (and the corresponding averages when capitalized). Additionally, besides the classical Reynolds number previously mentioned, turbulent channel flow can be better characterized by the friction Reynolds number

$$\text{Re}_\tau = \frac{u_\tau \delta}{\nu}, \quad (37)$$

where  $u_\tau$  is the friction velocity,  $u_\tau = \sqrt{\nu(\partial_z U)|_{z=0}}$ . Also using  $u_\tau$ , the following dimensionless variables can be constructed

$$z^+ = \frac{z\nu}{u_\tau}, \quad u^+ = \frac{u}{u_\tau}, \quad (38)$$

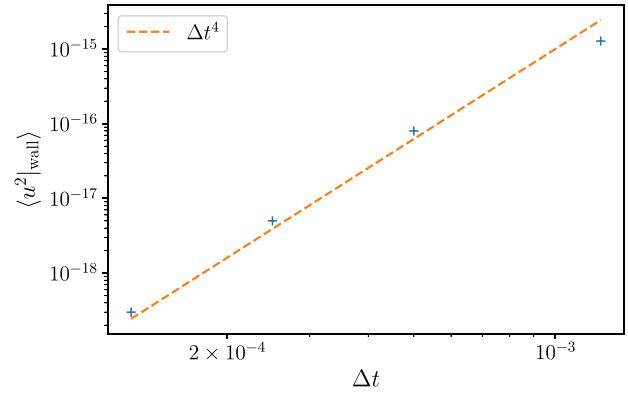
with which different results from experiments and simulations can be more directly compared.

Using the turbulent velocity fields, for each simulation we first compute the diagonal elements of the two-point correlation tensor

$$Q_{ij}(z, r_l) = \frac{\langle v'_i(\mathbf{x}) v'_j(\mathbf{x} + r_l \hat{\mathbf{r}}_l) \rangle_{x,y,t}}{\langle v'_i(\mathbf{x}) v'_j(\mathbf{x}) \rangle_{x,y,t}}, \quad (39)$$

where  $r_l$  is a spatial displacement, and  $\hat{\mathbf{r}}_l$  is the unit vector in the direction of the displacement. In the periodic directions the correlations  $Q_{ii}(z, r_l)$  are expected to decay to zero for separations  $r_l$  of, at most, half the box length. Indeed this can be observed in Fig. 6, where all the diagonal elements of the correlation tensor were plotted as a function of both  $r_x$  and  $r_y$ , and both near the bottom wall (for  $z^+ = 10$ ) and in the center of the channel (i.e., for  $z/\delta = 1$ ).

Expressed in dimensionless units, the mean profile for the streamwise velocity in a fully developed turbulent channel flow



**Fig. 4.** Mean squared streamwise velocity at the wall for simulation T1 as a function of the timestep  $\Delta t$  (blue marks). The scaling  $\Delta t^4$  is shown as a reference by the dashed line. The point to the right with the largest timestep has a CFL condition of 0.625. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

has long been proposed to follow a universal logarithmic law far from the walls (known as the von Kármán law), a fact that both experiments and simulations support [7]. This means that, for sufficiently large  $\text{Re}_\tau$ , for  $z^+ \gg 1$  and for  $z \ll \delta$ , the law

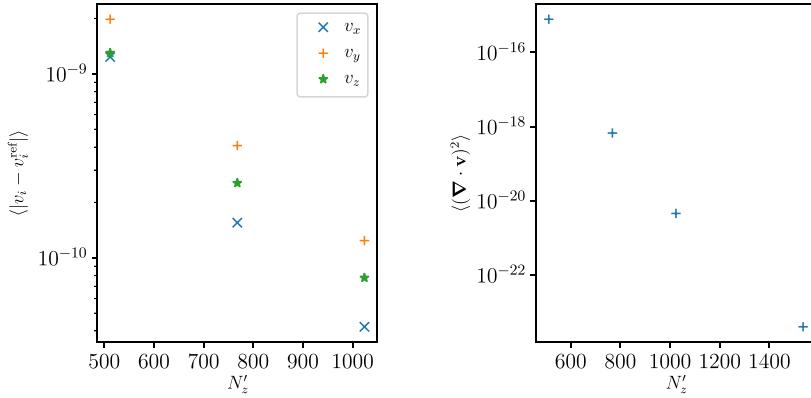
$$U^+ = \frac{1}{\kappa} \ln(z^+) + B, \quad (40)$$

should hold, where  $\kappa$  is the von Kármán constant. To verify our simulations are compatible with this law, we estimate the mean profiles for each of our turbulent simulations, shown in Fig. 7. For the case of T1 and T2 transitional effects are still dominant, and no intervals compatible with a logarithmic law are present. For T3 and T4, on the other hand, the mean profile is in reasonable agreement with the logarithmic law. A best fit of the data for T4 yields  $\kappa = 0.3993 \pm 0.0009$  and  $B = 5.41 \pm 0.02$ , values which are in agreement with previous studies [7,71] (the curve corresponding to these values is shown as a reference in Fig. 7). Moreover, for additional validation of the mean profile, we compare our results with publicly available data from a channel flow simulation at  $\text{Re}_\tau = 182$  performed by Lee and Moser [71] (abbreviated L2015 from now on), done at a resolution of  $1024 \times 192 \times 512$  grid points in a domain of size  $8\pi \times 2 \times 3\pi$  using a Fourier–Galerkin method in the periodic directions and a 7th order B-spline collocation in the wall-normal direction, which in their case is along the  $y$  axis. This data is available at [http://turbulence.oden.utexas.edu/channel2015/content/Data\\_2015\\_0180.html](http://turbulence.oden.utexas.edu/channel2015/content/Data_2015_0180.html). Their mean profile is also shown in Fig. 7. Even though for our simulations the aspect ratio of the box is somewhat different and the corresponding values of  $\text{Re}_\tau$  differ from those in L2015, for the case of run T4 ( $\text{Re}_\tau = 173$ ) a considerable interval of agreement in the mean profile is found.

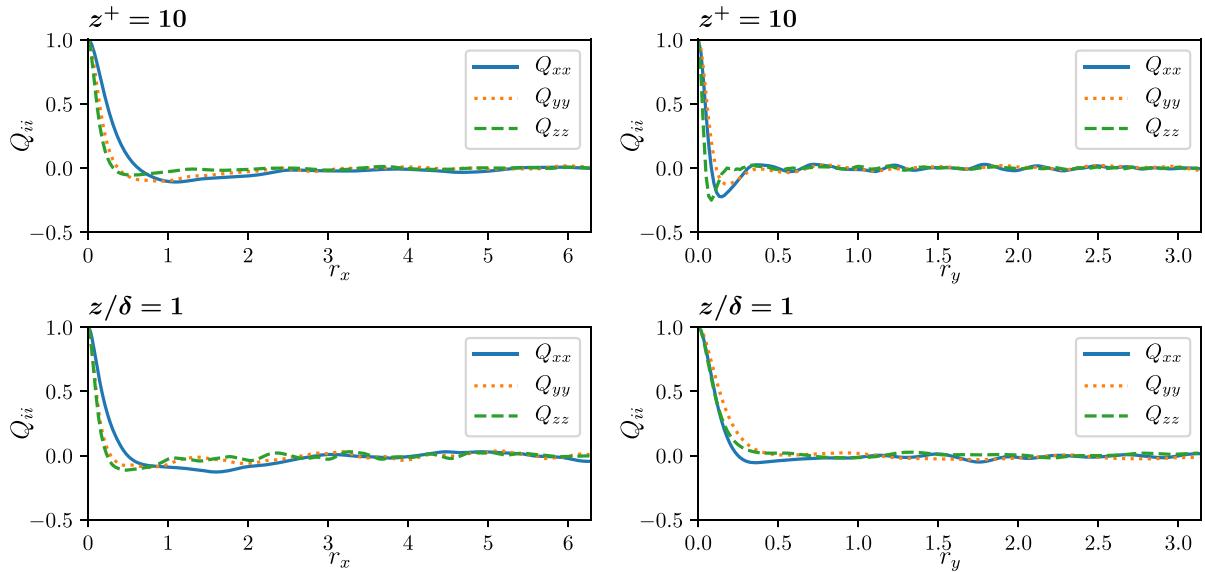
To further analyze the presence of a logarithmic law in our mean profiles, we calculate the so-called indicator function  $\beta$  [13], defined as

$$\beta(z^+) = z^+ \frac{\partial U^+}{\partial z^+}. \quad (41)$$

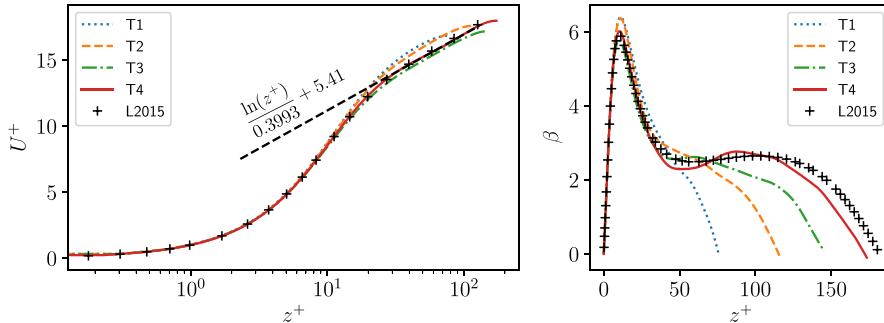
The indicator function is flat and equal to the von Kármán constant  $\kappa$  when a log-law is present. In Fig. 7 a plot of  $\beta(z^+)$  is shown for all the simulations. Although, in agreement with our previous observation, no horizontal interval is found for T1 and T2, an approximately flat interval is found for run T3 and an even larger plateau for T4, suggesting that a logarithmic law is indeed compatible with our results. Fig. 7 also shows a comparison of  $\beta$  with the data from L2015 at a larger value of  $\text{Re}_\tau$ , indicating a



**Fig. 5.** Left: Mean  $L_1$  error for each velocity field component as a function of the number of grid points  $N'_z$  (counting continuation points) in the vertical non-periodic direction, computed after evolving the steady-state turbulent solution T1 for one large-scale turnover time. All simulations have  $C_z = 33$  continuation points, so the actual vertical resolution is  $N_z = N'_z - 33$ . A simulation with  $N'_z = 1536$  is considered as the reference velocity field  $v^{\text{ref}}$ . The same time step and horizontal resolution is employed for all the simulations. Right: Mean squared divergence of the velocity field for the same set of simulations as a function of the number of vertical grid points  $N'_z$ .



**Fig. 6.** Diagonal elements of the two-point correlation tensor for run T4 as a function of streamwise (left) and spanwise (right) separations, near the wall at 10 wall units (top), and at the centerline (bottom).

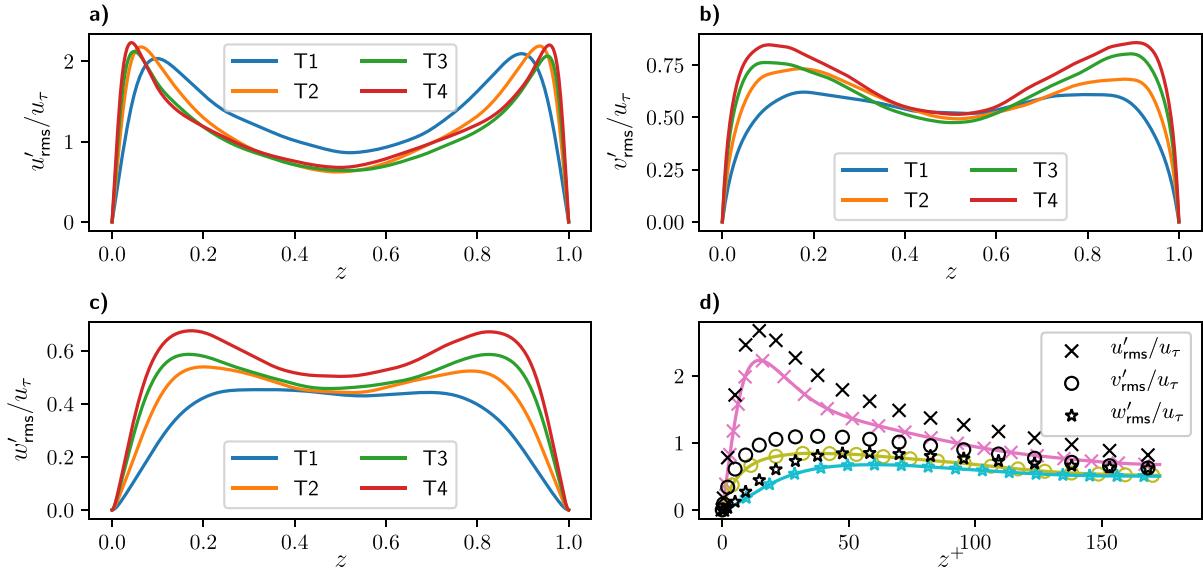


**Fig. 7.** Left: Mean vertical profile in wall coordinates of the streamwise velocity for turbulent runs T1 to T4. A logarithmic profile is shown as a reference by the dashed line. Right: Indicator function for runs T1 to T4. In both cases data for a simulation from L2015 is shown for comparison, and marked with black crosses.

good agreement for this quantity near the boundary layer (small values of  $z^+$ ), and the correct trend as  $\text{Re}_\tau$  is increased in the log-region.

In Fig. 8, the profiles for the root mean squared turbulent velocity normalized by the friction velocity,  $v_{i,\text{rms}}/u_\tau =$

$(\langle v_i'^2 \rangle_{x,y,t})^{1/2}/u_\tau$ , are shown. The first thing to notice is that for all turbulent velocity components and for all the simulations, a good degree of symmetry is observed, indicating that an adequate time span was considered for the computation of the mean quantities. Additionally, as expected, the maximum amplitude of



**Fig. 8.** Mean profiles of the root mean squared turbulent velocities normalized by  $u_\tau$  and as a function of  $z$  for (a)  $u'_\text{rms}$ , (b)  $v'_\text{rms}$ , and (c)  $w'_\text{rms}$ . (d) Comparison (for  $z$  in wall units) of the normalized r.m.s. turbulent velocities between simulation T4 (connected by solid lines) and L2015 (with unconnected symbols).

the streamwise turbulent fluctuations is located closer to the wall and presents greater values for larger Reynolds numbers. The centerline turbulent fluctuations, nonetheless, seem to decrease for higher  $Re_\tau$  but it soon reaches an approximately constant value. For the spanwise and wall normal components a monotonous increase of the turbulence intensity near the wall is observed conforming  $Re_\tau$  grows. When comparing the turbulent intensities for simulation T4 with those observed in L2015 (also in Fig. 8), a good level of agreement is obtained considering the different values of  $Re_\tau$  between simulations, even though the T4 run was evolved for a shorter period of time and the estimation of the mean amplitude can be affected by finite statistical sampling.

We also study the scale dependence of the turbulent fluctuations. To this purpose the 1D energy spectra of the turbulent velocity components (normalized by the friction velocity) are computed. These spectra are defined as

$$E_{v_i'v_i'}(k_x, z) = \sum_{k_y} |\hat{v}_i(k_x, k_y, z)|^2, \quad (42)$$

$$E_{v_i'v_i'}(k_y, z) = \sum_{k_x} |\hat{v}_i(k_x, k_y, z)|^2. \quad (43)$$

To enable a better comparison with the spectra obtained in L2015, which uses a different domain size, we express the energy spectra as a function of the dimensionless wavenumbers  $k_i^+ = k_i v / u_\tau$ . The result for simulation T4 is shown in Fig. 9 for two different heights, one near the height of maximum r.m.s. turbulent velocity at  $z^+ = 10$ , and the other at the center of the channel (i.e., at  $z = \delta$ ). A smoothly decaying spectra is obtained in all the cases, with the exception of the spanwise spectra near the wall, for which a small accumulation of energy at the largest wavenumbers can be seen, resulting from aliasing. The fast drop that follows in all spectra for larger wavenumbers is associated to our filter. When compared to the spectra from L2015, a significant level of agreement is observed, with very similar decay rates for all velocity components, and for all wavenumbers considered. Even more, some specific features are also reproduced, like the crossing between  $E_{u'u'}(k_x^+, 0.5)$  and  $E_{w'w'}(k_x^+, 0.5)$  at  $k_x^+ \approx 2 \times 10^{-2}$ , and the accumulation of energy present in  $E_{u'u'}(k_y^+)$  and  $E_{w'w'}(k_y^+)$  near the wall for intermediate wavenumbers.

Finally, Fig. 10 shows 3D renderings of the streamwise velocity and of the r.m.s. vorticity using the software VAPOR [72]. Note

the small instantaneous streamwise velocities (indicated by blue) near the boundaries, and the streaks in the streamwise direction. The generation of vorticity near the wall, and the formation of vortex tubes, is more clearly seen in the renderings of the r.m.s. vorticity, with the largest values of this quantity (indicated by red) taking place near the walls.

Overall, the results shown so far for channel flow simulations indicate our method can give solutions to this problem that are in good agreement with previous studies using pseudospectral or high order B-spline methods, while using a regular grid and (as a result) relatively larger time steps. Solution for the pressure in our method is also relatively straightforward, with a minimum overhead compared with pseudospectral methods in fully periodic domains. However, as already mentioned, channel flow simulations require sufficient resolution near the walls, and as a result we have used a large resolution in the vertical direction to properly resolve the boundary layer. In the next section we consider a problem for which a regular grid provides more numerical advantages.

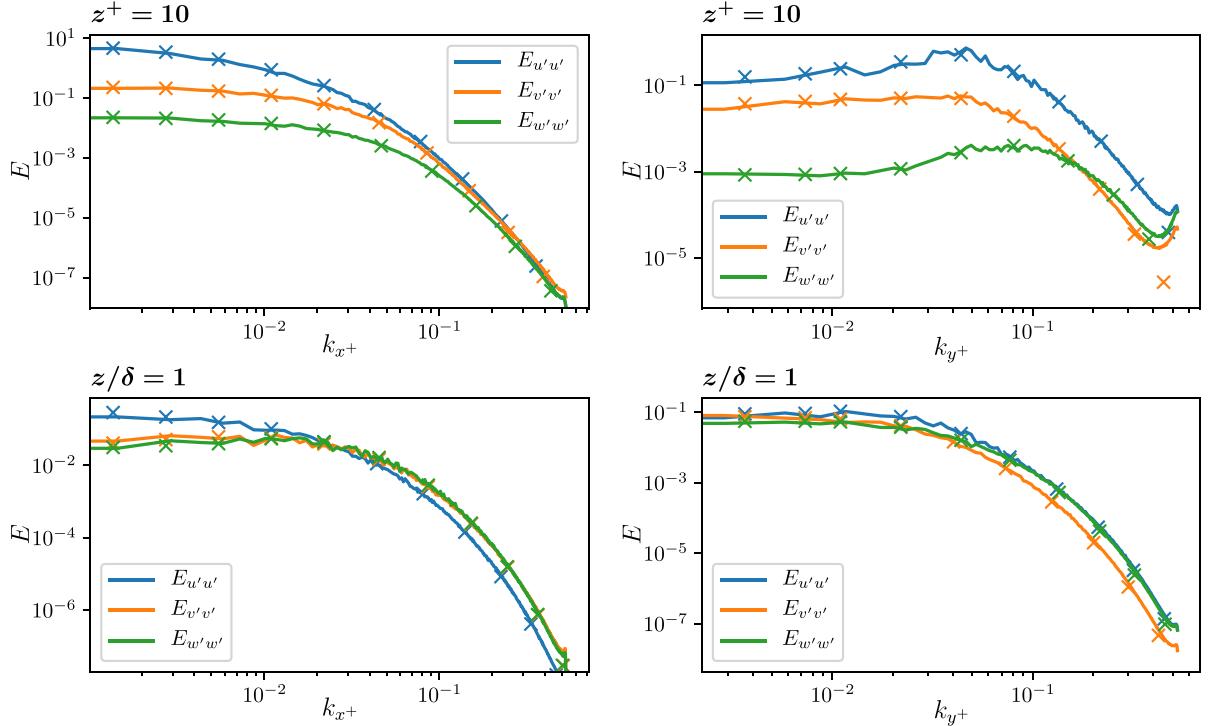
## 7. Plane Rayleigh–Bénard convection simulations

We now analyze the application of the method presented above to the case of another set of equations. To this purpose we study the turbulent plane Rayleigh–Bénard convection problem [17–20,73], where the space between two plates held at constant temperature is filled with a fluid. If the bottom plate has a fixed temperature  $T_b$  much greater than the temperature at the top  $T_t$ , the fluid destabilizes and convection develops, displaying the well known Bénard cells. Further increasing the temperature difference between the plates leads to turbulent convection [74]. For incompressible fluids, it is not unusual to study the case in which density variations with temperature are so small that they only affect the dynamics of the fluid via the buoyancy force, leading the incompressible Boussinesq equations

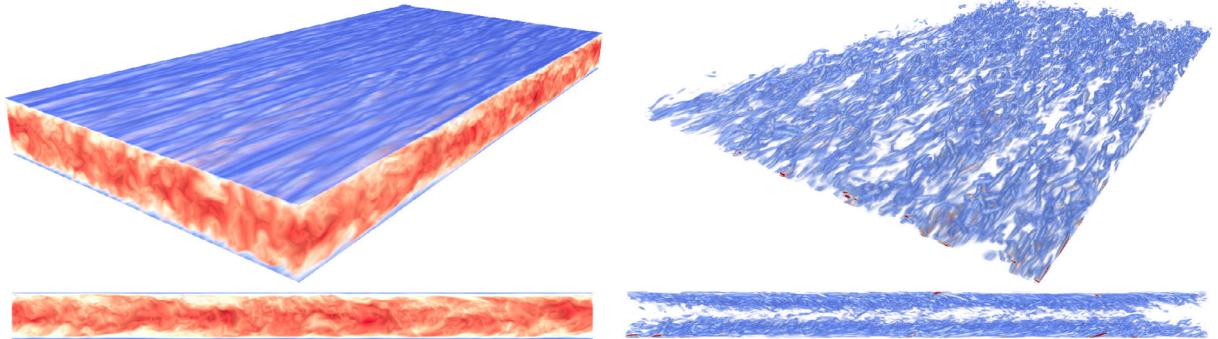
$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = -\nabla p + \gamma \theta \hat{\mathbf{z}} + \nu \nabla^2 \mathbf{v}, \quad (44)$$

$$\nabla \cdot \mathbf{v} = 0, \quad (45)$$

$$\frac{\partial \theta}{\partial t} + (\mathbf{v} \cdot \nabla) \theta = \gamma v_z + \kappa \nabla^2 \theta. \quad (46)$$



**Fig. 9.** Energy spectra  $E_{v'_i v'_i}$  for  $u'$ ,  $v'$ , and  $w'$ , as a function of the streamwise (left) and spanwise (right) wavenumbers, both near the wall (top, for  $z^+ = 10$ ) and at the centerline (bottom, for  $z/\delta = 1$ ) for simulation T4 (solid lines). All quantities are expressed in wall units. Spectra from L2015 is marked with the same colors but using unconnected symbols.



**Fig. 10.** 3D renderings of the streamwise velocity in the entire domain (left), and of the r.m.s. vorticity near the bottom wall (right) in the channel flow simulation T4. The renderings on the bottom show the same quantities from a side view. Blue corresponds to small values, red to high values of each quantity. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Here, the total temperature is  $T = T_0 + T'$ , where  $T_0$  is a linear background profile  $T_0(z) = T_b - z\Delta T/h$  with  $h = L_z$  the height of the domain,  $\Delta T = T_b - T_t$  the temperature difference between the plates (with  $T_b$  and  $T_t$  the temperatures at the bottom and top plates, respectively), and  $T'$  the “fluctuating” temperature that corrects the background profile. The total pressure is  $P = p_0 + p$ , with  $p$  the correction to a background hydrostatic pressure  $p_0 = \alpha g z (T_b - z\Delta T/h) + P_0$ , with  $\alpha$  the fluid thermal expansion coefficient,  $g$  the acceleration of gravity, and  $P_0$  a constant (as before, the mean mass density of the fluid is  $\rho = 1$  in dimensionless units). In Eqs. (44) and (46) we write the correction to the total temperature in units of velocity by defining  $\theta = T' \sqrt{\alpha g h / \Delta T}$ . Finally,  $\gamma = \sqrt{\alpha g \Delta T / h}$ , and  $\nu$  and  $\kappa$  are respectively the kinematic viscosity and the thermal diffusivity. With these choices the boundary conditions for  $\mathbf{v}$  and  $\theta$  are periodic in the  $x$  and  $y$  directions, while  $\mathbf{v} = \mathbf{0}$  and  $\theta = 0$  at both  $z = 0$  and  $z = h$ .

The system of Eqs. (44)–(46) has an important dimensionless number that controls the instability of the system to natural convection, the Rayleigh number defined as

$$\text{Ra} = \frac{\alpha g h^3 \Delta T}{\nu \kappa} = \frac{\gamma^2 h^4}{\nu \kappa}. \quad (47)$$

Other important dimensionless numbers are the Prandtl number,  $\text{Pr} = \nu/\kappa$ , which is 1 in all our simulations, and the Nusselt number which quantifies the ratio of convective to conductive heat transfer, defined in [73] as

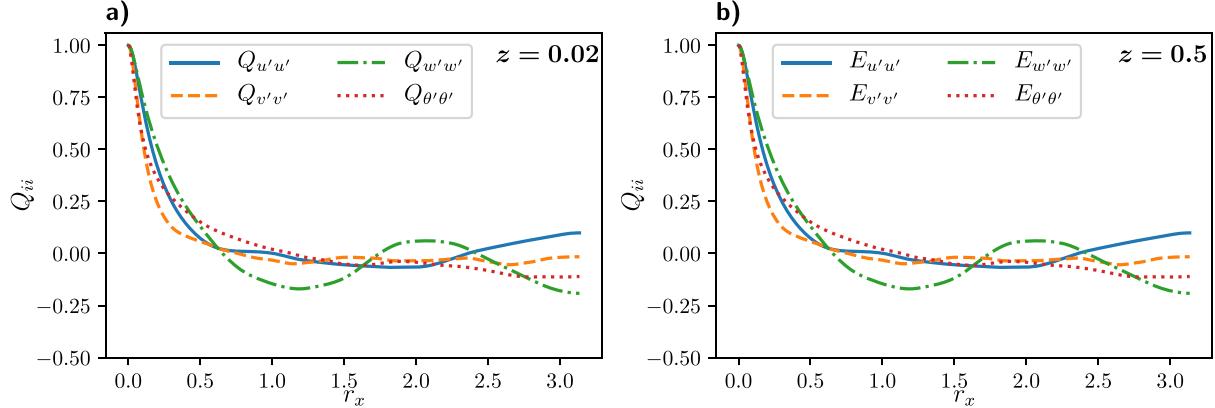
$$\text{Nu} = \frac{h}{\Delta T} \left. \frac{\partial T}{\partial z} \right|_{\text{wall}} = \frac{1}{\gamma} \left. \frac{\partial \theta}{\partial z} \right|_{\text{wall}} - 1. \quad (48)$$

Rayleigh–Bénard convection has been studied profusely in the literature, in different configurations and regimes [17–20,73]. Here we just consider a few simulations and typical quantifications of the flow, to illustrate how our numerical method can

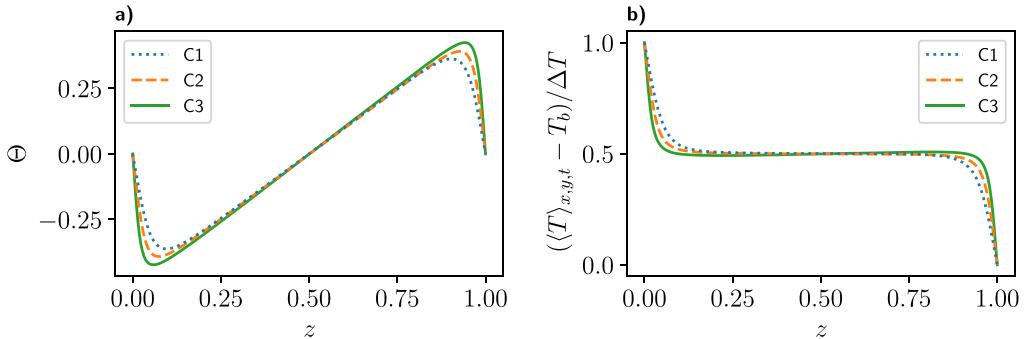
**Table 2**

Parameters of the Rayleigh–Bénard convection simulations. “Run” labels each simulation,  $N_x \times N_y \times N_z'$  gives the linear resolution in each direction, and  $C_z$  and  $d_z$  are the number of continuation and boundary matching points in the  $z$  direction, respectively. The kinematic viscosity and thermal diffusivity are  $\nu$  and  $\kappa$ , respectively. Finally,  $\gamma = \sqrt{(\alpha g \Delta T / h)}$  (where  $\alpha$  is the fluid thermal expansion coefficient,  $g$  the acceleration of gravity,  $\Delta T$  the temperature difference between the plates, and  $h$  the distance between the plates),  $\text{Ra}$  is the Rayleigh number, and  $\text{Nu}$  is the Nusselt number.

Run	$N_x \times N_y \times N_z$	$C_z$	$d_z$	$\nu$	$\kappa$	$\gamma$	$\text{Ra}$	$\text{Nu}$	$\text{NuRa}^{1/3}$
C1	$256 \times 256 \times 103$	25	5	$1 \times 10^{-3}$	$1 \times 10^{-3}$	1	$1 \times 10^6$	6.55	0.071
C2	$512 \times 512 \times 231$	25	5	$5 \times 10^{-4}$	$5 \times 10^{-4}$	1	$4 \times 10^6$	10.89	0.069
C3	$1024 \times 1024 \times 479$	33	7	$2.5 \times 10^{-4}$	$2.5 \times 10^{-4}$	1	$1.6 \times 10^7$	17.99	0.066



**Fig. 11.** Diagonal elements of the two point correlation tensor  $Q_{ii}$  for simulation C3 at (a): the thermal boundary layer  $z = \delta_\theta = 0.02$  and (b) the center of the box  $z = 0.5$ .



**Fig. 12.** (a): Mean temperature correction (in velocity units)  $\Theta$  as a function of the height  $z$ . The three simulations at different resolutions are shown (see inset for references). (b): Normalized mean total temperature profile  $(\langle T \rangle_{x,y,t} - T_b) / \Delta T$  as a function of the height  $z$ . Note how temperature becomes uniform in the center of the domain.

be used for a set of PDEs different from the simplest case of the incompressible Navier–Stokes equation. Using the SPECTER code we solve Eqs. (44)–(46) inside a box of size  $L_x \times L_y \times L_z = 2\pi \times 2\pi \times 1$ . Values of the spatial resolution, of the relevant parameters, and of the characteristic dimensionless numbers are given in Table 2. It is worth noting that a turbulent convection regime is attained in all the cases.

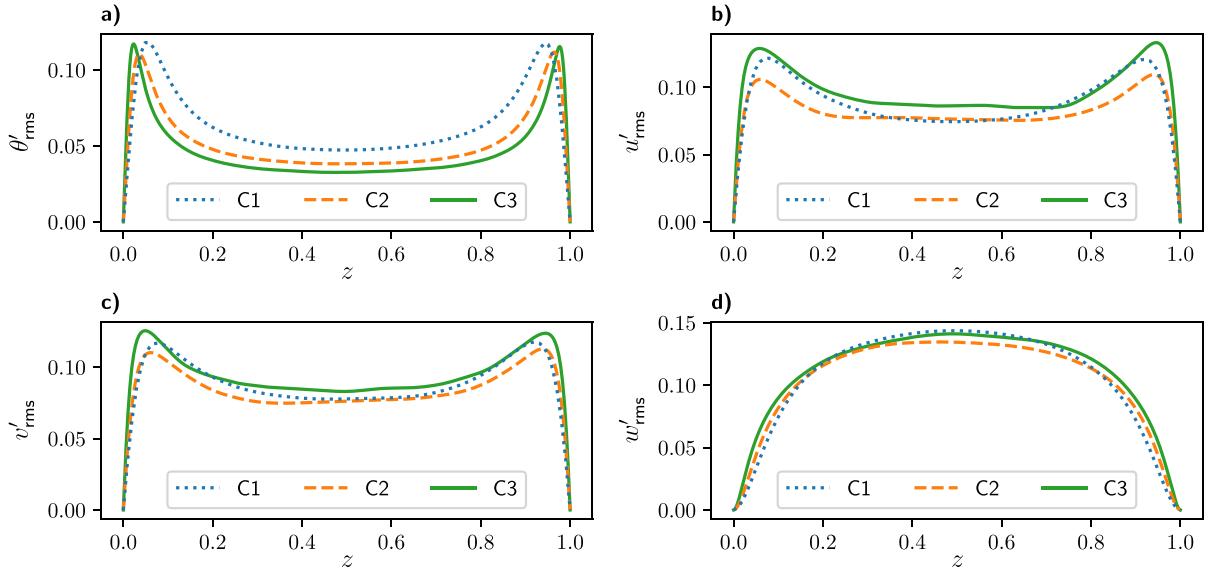
As for the channel flow, we start by computing the diagonal elements of the two-point correlation tensor. The entries associated with the velocity components are the same as those given in Eq. (39), while the temperature dependent elements are defined as

$$Q_{\theta j}(z, r_l) = Q_{j\theta}(z, r_l) = \frac{\langle \theta'(\mathbf{x}) v'_j(\mathbf{x} + r_l \hat{\mathbf{r}}_l) \rangle_{x,y,t}}{\langle \theta'(\mathbf{x}) v'_j(\mathbf{x}) \rangle_{x,y,t}}, \quad \text{for } j \in \{x, y, z\}, \quad (49)$$

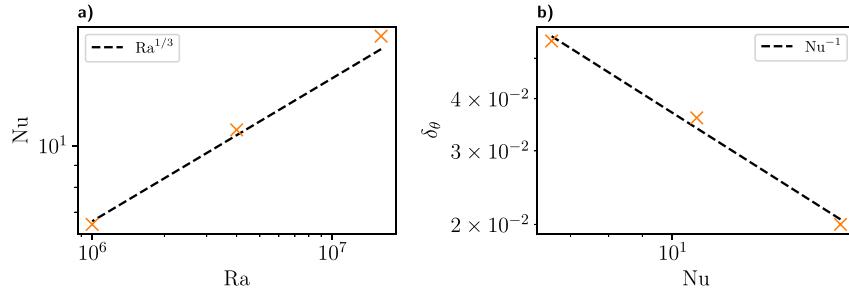
$$Q_{\theta\theta}(z, r_l) = \frac{\langle \theta'(\mathbf{x}) \theta'(\mathbf{x} + r_l \hat{\mathbf{r}}_l) \rangle_{x,y,t}}{\langle \theta'(\mathbf{x}) \theta'(\mathbf{x}) \rangle_{x,y,t}}. \quad (50)$$

As before, primed variables represent fluctuations, i.e., we decompose  $\theta = \Theta + \theta'$ , with capitalized variables denoting mean vertical profiles, i.e.,  $\Theta = \langle \theta \rangle_{x,y,t}$ . The result of the two-point correlations for simulation C3 is shown in Fig. 11 at two different heights, one near the height of maximum turbulent r.m.s. thermal fluctuations ( $z = 0.02$ ), and the other in the center of the domain. As for the channel flow, horizontal correlations decay to zero before the half box length, and then fluctuate around zero. However, for convection the two-point correlations decay slower than in the channel flow, in good agreement with the development of large-scale convective cells in the flow as will be shown in visualizations later.

Statistically, the convection process should remove heat from the hot plate and transport it to the cold plate. This can be verified by estimating the mean profile for the temperature correction  $\Theta$ , which should be negative near the bottom plate (i.e., colder than the non-convective solution), and positive near the top plate (i.e., hotter). In Fig. 12 we show  $\Theta$  for all our simulations. The results are compatible with this observation. Even more, as expected, increasing  $\text{Ra}$  (i.e., increasing convective action) results in



**Fig. 13.** Root mean square (r.m.s.) fluctuations of (a): the turbulent temperature fluctuation in velocity units  $\theta'_{\text{rms}}$ , (b): the turbulent velocity component in the  $x$  direction  $u'_{\text{rms}}$ , (c): the turbulent velocity component in the  $y$  direction  $v'_{\text{rms}}$ , and (d): the turbulent velocity component in the wall normal direction  $w'_{\text{rms}}$ . References for the three simulations are indicated in the insets.



**Fig. 14.** (a) Average Nusselt number  $\text{Nu}$  as a function of the Rayleigh number  $\text{Ra}$  for simulations C1, C2, and C3. A  $\text{Ra}^{1/3}$  scaling is shown by the dashed line as a reference. (b): Thermal boundary layer thickness  $\delta_\theta$  as a function of the Nusselt number  $\text{Nu}$  for simulations C1, C2, and C3, along with a  $\text{Nu}^{-1}$  dashed line for reference.

larger temperature corrections which tend to concentrate closer to the plates. In terms of the temperature correction in velocity units  $\theta$ , it is straightforward to recover the actual pointwise temperature (normalized by the temperature difference at the boundaries) as

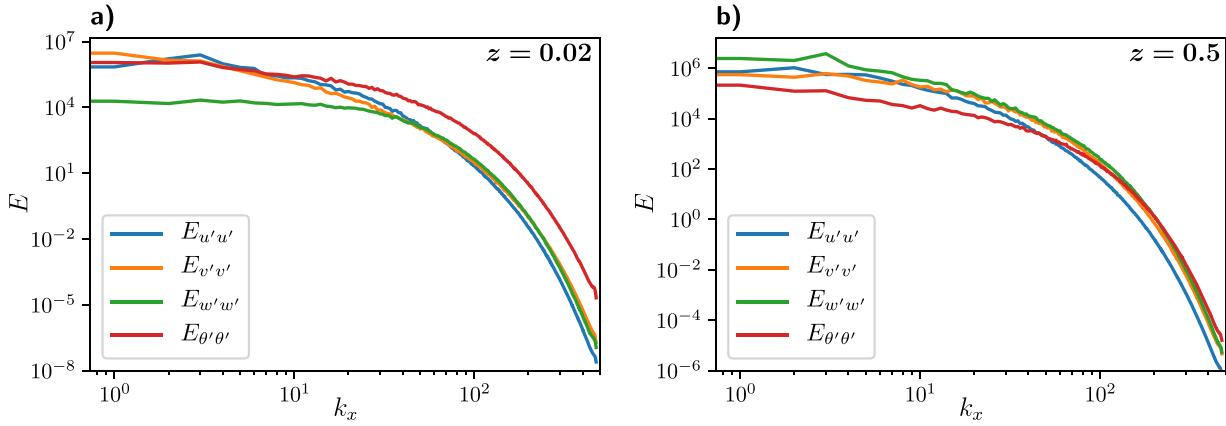
$$\frac{T(x, y, z) - T_b}{\Delta T} = \frac{\theta(x, y, z) - \gamma z}{\gamma h}. \quad (51)$$

Using this expression we can calculate the mean temperature profile  $(\langle T \rangle_{x,y,t} - T_b)/\Delta T$ , which is also shown in Fig. 12. The result is compatible with the profile reported in [75], in which temperature is approximately constant in most of the domain (i.e., convection mixes fluid elements resulting in homogeneous temperature) except in a small region near each plate associated to the thermal boundary layer. As expected, the latter region becomes narrower with increasing  $\text{Ra}$ .

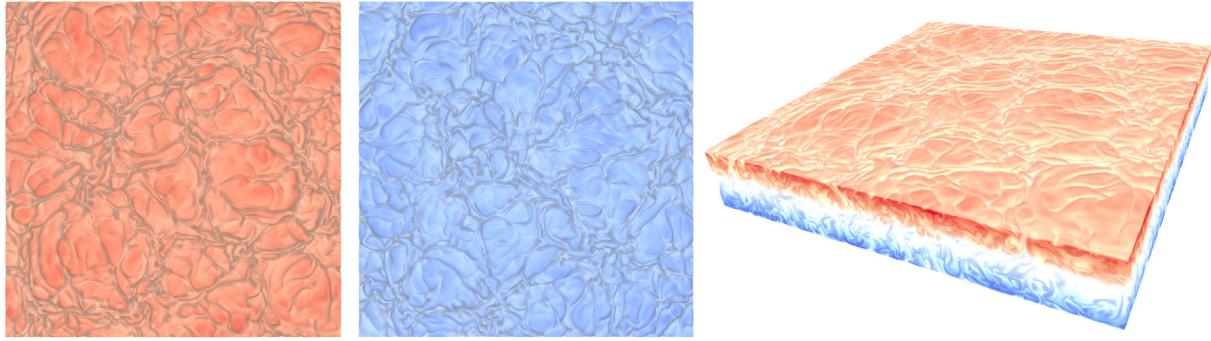
We now study the variation of r.m.s. fluctuations as a function of the height, both for the temperature correction and for the velocity. The results are shown in Fig. 13. The first thing to notice is the symmetry of the profiles, indicating an appropriate timespan for the computed averages, as well as the resemblance between the profiles for  $u'_{\text{rms}}$  and  $v'_{\text{rms}}$ , which is expected from the symmetry between  $x$  and  $y$  directions in the configuration. Additionally, for increasing values of  $\text{Ra}$  it is clearly seen that the maximum r.m.s. fluctuations of  $\theta'_{\text{rms}}$ ,  $u'_{\text{rms}}$  and  $v'_{\text{rms}}$  tend to concentrate closer to the plates. The increase in the amplitude of their

maxima as a function of  $\text{Ra}$  is, however, mild, in agreement with results in [19]. Another important thing to notice is that all  $\theta'$ ,  $u'$ , and  $v'$  attain their maximum r.m.s. values near the wall, whereas  $w'$  is considerably smaller in the same region, which is consistent with results in [73]. On the contrary, it is in the center of the box where  $w'$  attains its maximum r.m.s. amplitude, where it becomes the most relevant quantity. The opposite behavior is found when analyzing  $\theta'$ , as its r.m.s. value considerably drops in the center of the box and its centerline value decreases monotonously with  $\text{Ra}$ , as reported in [19].

As a way to compare our results with previous simulations of convection, we look at the scaling of the Nusselt number with the Rayleigh number, and at the scaling of the thickness of the thermal boundary layer. Many studies of Rayleigh-Bénard turbulent convection reported a dependence of  $\text{Nu}$  (which usually cannot be directly controlled and is estimated *a posteriori*) with  $\text{Ra}$  (the control parameter). In simulations in [19] it was found a scaling  $\text{Nu} \sim \text{Ra}^{1/3}$ , with the product  $\text{NuRa}^{1/3}$  being approximately constant for  $\text{Ra}$  ranging from  $10^6$  to  $10^{14}$ . To test these results in our simulations, we estimate the mean Nusselt number using Eq. (48). The results are shown in Table 2 and in Fig. 14, where  $\text{Nu}$  is shown as a function of  $\text{Ra}$ , with a  $\text{Ra}^{1/3}$  scaling law indicated for comparison. The results are compatible with those in [19]. Even more, the product  $\text{NuRa}^{1/3}$  is in the range 0.066–0.071 for all our simulations (see Table 2), in agreement with experimental results in [76] for convection at  $\text{Pr} \approx 1$ . The scaling of the thermal



**Fig. 15.** Turbulent energy spectra as a function of the  $k_x$  wavenumber for  $u'_{\text{rms}}$ ,  $v'_{\text{rms}}$ ,  $w'_{\text{rms}}$ , and  $\theta'_{\text{rms}}$  at (a): the thermal boundary layer thickness  $z = 0.02$ , and (b): the center of the domain  $z = 0.5$  (see insets for references).



**Fig. 16.** 3D rendering of the temperature correction  $\theta$  in run C3. Note the total temperature includes the background temperature profile, which is not shown here. From left to right: view of the top of the domain, view from the bottom, and side view. Note hot fluid (in red) is transported to the top while cold fluid (in blue) is transported to the bottom. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

boundary layer  $\delta_\theta$  as a function of  $\text{Nu}$  is predicted in [73,77] to be  $\delta_\theta \sim \text{Nu}^{-1}$ . We tested this scaling in our simulations by estimating  $\delta_\theta$  as the height for which  $\theta_{\text{rms}}$  attained its maximum (we verified that other possible definitions for the width of the thermal boundary layer give similar results). In Fig. 14 we show  $\delta_\theta$  as a function of  $\text{Nu}$ , with a  $\text{Nu}^{-1}$  scaling law for reference. Our results are in agreement with the aforementioned scaling.

As before, we also study the scale dependence of the turbulent fluctuations utilizing the 1D energy spectra, defined for the velocity field in the same way as Eqs. (42) and (43) and for  $\theta$ , similarly as

$$E_{\theta'\theta'}(k_x, z) = \sum_{k_y} \left| \hat{\theta}'(k_x, k_y, z) \right|^2, \quad (52)$$

$$E_{\theta'\theta'}(k_y, z) = \sum_{k_x} \left| \hat{\theta}'(k_x, k_y, z) \right|^2. \quad (53)$$

The resulting spectra are shown in Fig. 15 as a function of  $k_x$  (a similar result is obtained for the spectra as a function of  $k_y$ ) for simulation C3 at two different heights: one at  $z = 0.02 \approx \delta_\theta$ , and the other at the center of the box. The first thing to notice is that smooth spectra without notable aliasing are obtained in all cases. Another interesting feature is the absence of a sharp peak for low wavenumbers (i.e., no characteristic roll lengthscale), indicating that the attained regime is indeed of turbulent Rayleigh–Bénard convection. As noted before in Fig. 13, near the wall and for low wavenumbers,  $\theta'$ ,  $u'$ , and  $v'$  are approximately equal in amplitude, whereas the vertical velocity  $w'$  is considerably smaller. This feature is not present for larger wavenumbers, where the three velocity components have approximately the same power and

$\theta'$  has significant more power, indicating the presence of very small-scale structures in latter field. The behavior at the center of the box is considerably different. For low wavenumbers it is the vertical velocity  $w'$  that contains most of the energy, while  $\theta'$  has a smaller power than any velocity component, both facts also in agreement with the features found in Fig. 13. As the wavenumber increases, nonetheless, the energy becomes more equally distributed between  $\theta'$  and the velocity components.

Finally, Fig. 16 shows a 3D rendering of the temperature correction  $\theta$  using the software VAPOR. Note the total temperature is given by  $\theta\sqrt{\Delta T/(\alpha gh)}$  plus the background temperature profile, and as a result the renderings only show how convection corrects the background temperature. As expected, hot fluid is transported to the top while cold fluid is transported to the bottom. The multi-scale formation of convective cells, and of turbulent plumes near the walls [72], can be also clearly seen.

## 8. Conclusions

We presented a Fourier Continuation-based parallel pseudospectral method for incompressible fluids in cuboid non-periodic domains. The method produces dispersionless and dissipationless spatial derivatives with fast spectral convergence inside the domain, and with high order convergence at the boundaries. Thus, the method has no spurious dispersion, or “pollution”, errors that commonly arise in finite differences of finite elements methods. Incompressibility is imposed by solving a Poisson equation for the pressure. As the method is Fourier-based, the Laplace operator for this problem has a diagonal representation in spectral space, and is well-behaved and easy to invert. As a result,

solutions of the Poisson equation for the pressure are fast and computationally inexpensive, with the only overhead (compared with pseudospectral methods in periodic domains) of requiring computation of a homogeneous solution to satisfy the required boundary conditions. However, this homogeneous solution can be found analytically and thus generates a minimal overhead.

Being Fourier-based, the method also allows for fast estimation of spectral transforms in  $\mathcal{O}(N_i \log N_i)$  operations in each spatial direction  $i$ , even when boundary conditions are not periodic. It is compatible with uniform grids (although refined or nested meshes can also be implemented by splitting the domain in multiple subdomains, and matching boundary conditions between these subdomains). But in the case of uniform grids presented here, it allows for explicit time integration with a mild Courant–Friedrichs–Lewy (CFL) condition dominated by the advection term for sufficiently high Reynolds numbers, and thus with a time step for stability that scales linearly with the spatial resolution.

We also presented two time stepping methods, using a time-splitting technique to allow for independent imposition of the boundary conditions for the velocity field (or other fields in the PDEs considered), and for the pressure. A time-splitting forward Euler method was presented that has global error of order  $\mathcal{O}(\Delta t^2)$  and that satisfies the boundary conditions with error  $\mathcal{O}(\Delta t)$ , and we also presented a time-splitting low-storage  $o$ th order Runge–Kutta method that has both global and boundary condition errors of  $\mathcal{O}(\Delta t^o)$ .

The method with the time-splitting Runge–Kutta time evolving scheme was implemented in a publicly available code (SPECTER), and we briefly described efficient methods for its parallelization. This implementation of the method was validated against two problems with non-periodic boundary conditions: channel flow, and plane Rayleigh–Bénard convection under the Boussinesq approximation. For channel flows, we also compared our results with previous simulations using other high-order numerical methods. In both cases the method yields results compatible with previous studies.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

The authors acknowledge support from CONICET and ANPCyT through PIP, Argentina Grant No. 11220150100324CO and PICT, Argentina Grant No. 2015-3530. This work was also supported by National Science Foundation, USA and AFOSR, USA through contracts DMS-1714169 and FA9550-15-1-0043, and by the NSSEFF Vannevar Bush Fellowship, USA under contract number N00014-16-1-2808. We also thank the Physics Department at the University of Buenos Aires for providing computing time on its DIRAC cluster.

## Appendix A. Generation of random solenoidal 3D vector fields

In order to generate 3D incompressible noise that satisfies the no-slip condition to perturb the velocity field, we use a basis of eigenfunctions of the curl operator introduced in [78]. These eigenfunctions are obtained from the scalar potentials

$$\phi(x, y, z) = \sum_{nlm} a_{nlm} \exp[i(k_{x,n}x + k_{y,l}y)] g_m(z'), \quad (\text{A.1})$$

$$\psi(x, y, z) = \sum_{nlm} b_{nlm} \exp[i(k_{x,n}x + k_{y,l}y)] \sin \left[ m\pi \left( z' + \frac{1}{2} \right) \right]. \quad (\text{A.2})$$

by means of the toroidal–poloidal decomposition  $\mathbf{q} = \nabla \times (\nabla \times \phi \hat{\mathbf{z}}) + \nabla \times \psi \hat{\mathbf{z}}$ . The functions  $\mathbf{q}$  are eigenfunctions of the curl, and thus generate incompressible flows. Here,  $z'$  is the result of mapping  $z$  into the  $[-1/2, 1/2]$  domain, i.e.,  $z' = z/L_z - 1/2$ , and  $g_m(z')$  are the Chandrasekhar–Reid eigenfunctions (see pp. 634–637 of Ref. [79]).

$$g_m(z') = \begin{cases} \frac{\cosh(\lambda_m z')}{\cosh(\lambda_m/2)} - \frac{\cos(\lambda_m z')}{\cos(\lambda_m/2)} & \text{for } m \text{ odd,} \\ \frac{\sinh(\lambda_m z')}{\sinh(\lambda_m/2)} - \frac{\sin(\lambda_m z')}{\sin(\lambda_m/2)} & \text{for } m \text{ even,} \end{cases} \quad (\text{A.3})$$

where the sequence  $\lambda_m$  is constructed from the condition  $g'_m(-1/2) = g'_m(1/2) = 0$ .

Using Eqs. (A.1) and (A.2) we can construct a random and incompressible 3D vector by generating a superposition of the  $\mathbf{q}$  functions with random phases and amplitudes for both the  $a_{nlm}$  and  $b_{nlm}$  coefficients in the range  $4 \leq k_{x,n}^2 + k_{y,m}^2 \leq 25$  and  $2 \leq m \leq 5$ . The amplitudes decay as  $k_{nlm}^{-4}$  for  $\phi$  and  $k_{nlm}^{-3}$  for  $\psi$ , with  $k_{nlm} = (k_{x,n}^2 + k_{y,m}^2 + \lambda_m^2)^{1/2}$ . This generates a random 3D incompressible vector field that decays as  $k^{-2}$ , and whose poloidal and toroidal components are approximately balanced. Note that other choices for the decay of the spectrum of the perturbation can be easily obtained with other choices for the decay of the  $a_{nlm}$  and  $b_{nlm}$  coefficients.

## Appendix B. Mathematical aspects of the FC-gram Fourier continuation method

As mentioned in Section 3, and illustrated in Fig. 2, the Fourier continuation (FC) method utilizes a certain periodic-extension approach, the FC-Gram method, to produce rapidly-convergent Fourier series representations of non-periodic functions defined on one-dimensional intervals. Thus, for a given function  $f$ , which, without loss of generality, we assume is defined in the interval  $[0, 1]$ ,

$$f : [0, 1] \rightarrow \mathbb{R},$$

the FC method produces a  $b$ -periodic function

$$f^c : [0, b] \rightarrow \mathbb{R} \quad (b > 1),$$

defined on the interval  $[0, b] \supset [0, 1]$ , which closely approximates  $f(x)$  throughout the original interval  $[0, 1]$ —up to and including the endpoints 0 and 1.

Following [55, Sec. 3.1], more precisely, given a column vector  $\mathbf{f} = (f_0, \dots, f_{N-1})^T$  containing point-values of the function  $f$  on the equispaced grid  $0 = x_0 < x_1 < \dots < x_{N-1} = 1$ ,  $f_i = f(x_i)$ , the FC-Gram method [43,45,55] uses a subset of the given function values on small numbers  $d_\ell$  and  $d_r$  of matching points  $\{x_0, \dots, x_{d_\ell-1}\}$  and  $\{x_{N-d_r}, \dots, x_{N-1}\}$  contained in small subintervals on the left and right ends of the interval  $[0, 1]$  (of lengths  $\delta_\ell = (d_\ell - 1)\Delta x$  and  $\delta_r = (d_r - 1)\Delta x$ , where  $\Delta x$  is the distance between matching points) to produce, at first, a discrete periodic extension. Use of different numbers of matching points  $d_\ell$  and  $d_r$ ,  $d_\ell \neq d_r$  is desirable, for example, in cases in which one of the interval endpoints corresponds to a point on the boundary of a computational domain  $\Omega$  used for a PDE solution, while the other corresponds to a point interior to  $\Omega$ —at which the numerical solution is more accurate and whose error is smoother; see, e.g., [43]. Throughout this paper we have used the values  $d_\ell = d_r = d$  with  $d = 5$  and  $d = 7$ .

In order to obtain the desired discrete periodic expansion, the FC-Gram algorithm appends a number  $C$  of continuation function values in the interval  $[1, b]$  to the existing function data, so that the extension transitions smoothly from  $f_{N-1}$  back to  $f_0$ , as depicted in Fig. 2. The resulting vector  $\mathbf{f}^c$  can be viewed as a discrete set of values of a smooth and periodic function which is suitable for high-order approximation by means of the FFT algorithm in an interval of length  $(N + C)\Delta x$ . The  $C$  continuation values are produced on the basis of the discrete function defined by the vector  $\mathbf{f}$  together with a translation of it by a distance  $b$ . In detail, defining the sets  $\mathcal{D}_\ell = \{b + x_0, b + x_1, \dots, b + x_{d_{\ell-1}}\}$  and  $\mathcal{D}_r = \{x_{N-d_r}, x_{N-(d_r-1)}, \dots, x_{N-1}\}$ , the additional  $C$  needed values in the interval  $[1, b]$  are obtained as point values of an auxiliary trigonometric polynomial of periodicity interval  $[1 - \delta_r, 2b - (1 - \delta_r)]$  (with appropriately selected bandwidth) which closely approximates the function values on  $\mathcal{D}_r \cup \mathcal{D}_\ell$ . This approximating trigonometric polynomial is obtained as the result of a two-step process, namely: (1) Projection onto bases of orthogonal polynomials (Gram bases), and (2) continuation through use of a precomputed set of continuations-to-zero of each Gram polynomial, as explained in what follows.

The polynomial projection mentioned in step (1) above for the function values on  $\mathcal{D}_r$  and  $\mathcal{D}_\ell$  (cf. [45]) relies on use of a basis  $\mathcal{B}_r$  (resp.  $\mathcal{B}_\ell$ ), called the Gram basis, of the space of polynomials of degree  $< d_r$  (resp.  $d_\ell$ ) on the interval  $[1 - \delta_r, 1]$  (resp.  $[b, b + \delta_\ell]$ ) which is orthonormal with respect to the discrete scalar product  $(\cdot, \cdot)_r$  (resp.  $(\cdot, \cdot)_\ell$ ) defined by the discretization points  $\mathcal{D}_r$  (resp.  $\mathcal{D}_\ell$ ):

$$(g, h)_r = \sum_{x_i \in \mathcal{D}_r} g(x_i)h(x_i), \quad (B.1)$$

with a similar definition for  $(g, h)_\ell$ . The values of the resulting orthogonal polynomials at the discretization points in  $\mathcal{D}_r$  (resp.  $\mathcal{D}_\ell$ ) can be easily obtained by evaluating the QR factorization of the corresponding Vandermonde matrix. In view of the orthogonality property of the Gram polynomials, any given function can easily be projected onto the polynomial space directly via scalar product with each one of the orthogonal polynomials—for which, conveniently, only the function values at the Cartesian discretization points are required. Thus, the use of the Gram polynomial basis makes it easy to produce highly accurate approximations of various orders  $r$  of accuracy, by polynomials  $p_r$  whose values can be explicitly computed by means of certain well-conditioned linear algebra operations.

The algorithm also utilizes precomputed extensions, one for each polynomial in the Gram basis, into a smooth function defined for  $x \geq 1 - \delta_r$  which approximates  $p_r$  closely in the matching interval  $[1 - \delta_r, 1]$ , and which blends smoothly to zero for  $x \geq b$ . The rightward extensions, for example, are constructed as appropriately oversampled least squares approximations by Fourier series of periodicity interval  $[1 - \delta_r, 2b - (1 - \delta_r)]$ . Utilizing such smooth blending functions the algorithm proceeds to step (2): The evaluation of an extension from the function values at the set of points  $\mathcal{D}_r \cup \mathcal{D}_\ell$ . This is achieved, simply, by projection of the given set of function values onto the polynomial basis, followed by extension via the aforementioned rightward and leftward extension of Gram polynomials. With the extension in hand, an application of the discrete Fourier transform on the interval  $[0, b]$  to the vector of function values  $\mathbf{f}$  augmented by the  $C$  “continuation” values yields the desired trigonometric polynomial

$$f^c(x) = \sum_{k=-M}^M a_k e^{(2\pi i k x)/b} \quad \text{s.t. } f^c(x_i) = f(x_i), \quad i = 0, \dots, N-1. \quad (B.2)$$

For efficiency, the discrete Fourier transform is implemented by means of the Fast Fourier Transform (FFT).

The resulting continuation operation can be expressed in a block matrix form as

$$\mathbf{f}^c = \begin{bmatrix} I \\ A \end{bmatrix} \mathbf{f} = \begin{bmatrix} \mathbf{f} \\ A\mathbf{f} \end{bmatrix}, \quad (B.3)$$

where  $\mathbf{f}^c$  is a vector of the  $N+C$  continued function values,  $I$  is the  $N \times N$  identity matrix and  $A$  is the matrix containing the blend-to-zero continuation information. Defining the vector of matching points for the left and right as

$$\mathbf{f}_\ell = (f_0, f_1, \dots, f_{d_\ell-1})^T, \quad \mathbf{f}_r = (f_{N-d_r}, f_{N-d_r+1}, \dots, f_{N-1})^T, \quad (B.4)$$

the matrix  $A$  can be expressed in the form

$$A\mathbf{f} = A_\ell Q_\ell^T \mathbf{f}_\ell + A_r Q_r^T \mathbf{f}_r, \quad (B.5)$$

where the columns of  $Q_\ell$  and  $Q_r$  contain the  $d_\ell, d_r$  point values of each element of the corresponding Gram polynomial basis, and where the columns of  $A_\ell$  and  $A_r$  contain the corresponding  $C$  values that blend the polynomials in the left and the right Gram bases to zero. This step, which is responsible for all the ill conditioning in the continuation problem, can advantageously be performed as a precomputed operation, in high-precision accuracy, to produce a small re-usable set of parameters (a set of numbers proportional to both the number of Gram polynomials and extension points used—e.g., 231 numbers in our case), which can be utilized for continuation of arbitrary functions with negligible ill conditioning; full details in these regards may be found, e.g., in [55, Sec. 3.1].

## References

- [1] P.K. Yeung, X.M. Zhai, K.R. Sreenivasan, Proc. Natl. Acad. Sci. 112 (41) (2015) 12633–12638.
- [2] S. Khurshid, D.A. Donzis, K. Sreenivasan, Phys. Rev. Fluids 3 (8) (2018) 082601, <http://dx.doi.org/10.1103/PhysRevFluids.3.082601>.
- [3] S.A. Orszag, Phys. Fluids (ISSN: 00319171) 12 (12) (1969) II–250, <http://dx.doi.org/10.1063/1.1692445>.
- [4] S.A. Orszag, G.S. Patterson, Phys. Rev. Lett. (ISSN: 00319007) 28 (2) (1972) 76–79, <http://dx.doi.org/10.1103/PhysRevLett.28.76>.
- [5] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods in Fluid Dynamics, Springer Berlin Heidelberg, Berlin, Heidelberg, 1988, <http://link.springer.com/10.1007/978-3-642-84108-8>.
- [6] B. Fornberg, A Practical Guide to Pseudospectral Methods, in: Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, ISBN: 0521645646, 1998.
- [7] S.B. Pope, Turbulent Flows, Cambridge University Press, 2000, <http://dx.doi.org/10.1017/CBO9780511840531>.
- [8] C. Hirsch, Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics, Butterworth-Heinemann, ISBN: 0750665947, 2007.
- [9] J. Kim, P. Moin, R. Moser, J. Fluid Mech. (ISSN: 14697645) 177 (1987) (1987) 133–166, <http://dx.doi.org/10.1017/S0022112087000892>.
- [10] M.V. Zagarac, A.J. Smits, J. Fluid Mech. 373 (1998) 33–79, <http://dx.doi.org/10.1017/S0022112098002419>.
- [11] B. Eckhardt, T.M. Schneider, B. Hof, J. Westerweel, Annu. Rev. Fluid Mech. 39 (1) (2007) 447–468, <http://dx.doi.org/10.1146/annurev.fluid.39.050905.110308>.
- [12] K. Avila, D. Moxey, A. De Lozar, M. Avila, D. Barkley, B. Hof, Science (ISSN: 00368075) 333 (6039) (2011) 192–196, <http://dx.doi.org/10.1126/science.1203223>.
- [13] M. Lee, R.D. Moser, J. Fluid Mech. (ISSN: 14697645) 774 (2015) 395–415, <http://dx.doi.org/10.1017/jfm.2015.268>.
- [14] S.A. Orszag, L.C. Kells, J. Fluid Mech. (ISSN: 14697645) 96 (1) (1980) 159–205, <http://dx.doi.org/10.1017/S0022112080002066>.
- [15] M. Salewski, B. Eckhardt, Phys. Fluids (ISSN: 10897666) 27 (4) (2015) 1–17, <http://dx.doi.org/10.1063/1.4917281>.
- [16] L. Klotz, J.E. Wesfreid, J. Fluid Mech. (ISSN: 14697645) 829 (2017) 1–13, <http://dx.doi.org/10.1017/jfm.2017.614>.
- [17] S. Biringen, L.J. Peltier, Phys. Fluids A (ISSN: 08998213) 2 (5) (1990) 754–764, <http://dx.doi.org/10.1063/1.857729>.
- [18] E. Bodenschatz, W. Pesch, G. Ahlers, Annu. Rev. Fluid Mech. 32 (1) (2000) 709–778.
- [19] G. Amati, K. Koal, F. Massaioli, K.R. Sreenivasan, R. Verzicco, Phys. Fluids (ISSN: 10706631) 17 (12) (2005) 1–4, <http://dx.doi.org/10.1063/1.2140023>.
- [20] D. Lohse, K.-Q. Xia, Annu. Rev. Fluid Mech. 42 (1) (2010) 335–364.

[21] P. Moresco, T. Alboussière, J. Fluid Mech. 504 (2004) 167–181, <http://dx.doi.org/10.1017/S0022112004007992>.

[22] D. Krasnov, O. Zikanov, T. Boeck, J. Fluid Mech. 704 (2012) 421–446, <http://dx.doi.org/10.1017/jfm.2012.256>.

[23] S.K. Lele, J. Comput. Phys. (ISSN: 0021-9991) 103 (1) (1992) 16–42, [http://dx.doi.org/10.1016/0021-9991\(92\)90324-R](http://dx.doi.org/10.1016/0021-9991(92)90324-R).

[24] D. Gottlieb, S.A. Orszag, Numerical Analysis of Spectral Methods, Society for Industrial and Applied Mathematics, 1977, <http://dx.doi.org/10.1137/1.9781611970425>.

[25] X. Shan, D. Montgomery, H. Chen, Phys. Rev. A 44 (10) (1991) 6800–6818.

[26] P.D. Mininni, D.C. Montgomery, Phys. Fluids 18 (11) (2006) 116602.

[27] M. Fontana, P.D. Mininni, P. Dmitruk, Phys. Rev. Fluids 3 (12) (2018).

[28] K. Julien, M. Watson, J. Comput. Phys. 228 (5) (2009) 1480–1503.

[29] O. Botella, K. Shariff, Int. J. Comput. Fluid Dyn. 17 (2) (2003) 133–149.

[30] W.Y. Kwok, R.D. Moser, J. Jiménez, J. Comput. Phys. (ISSN: 0021-9991) 174 (2) (2001) 510–551, <http://dx.doi.org/10.1006/jcph.2001.6919>.

[31] A.T. Patera, J. Comput. Phys. 54 (3) (1984) 468–488.

[32] D. Rosenberg, A. Fournier, P. Fischer, A. Pouquet, J. Comput. Phys. 215 (1) (2006) 59–80.

[33] D. Rosenberg, A. Pouquet, P.D. Mininni, New J. Phys. (ISSN: 13672630) 9 (2007) <http://dx.doi.org/10.1088/1367-2630/9/8/304>.

[34] M.O. Deville, P.F. Fischer, E.H. Mund, High-Order Methods for Incompressible Fluid Flow, Cambridge University Press, 2002.

[35] L. Ying, G. Biros, D. Zorin, J. Comput. Phys. (ISSN: 0021-9991) 219 (1) (2006) 247–275, <http://dx.doi.org/10.1016/j.jcp.2006.03.021>.

[36] O. Bruno, T. Elling, C. Turc, Internat. J. Numer. Methods Engrg. 91 (10) (2012) 1045–1072, <http://dx.doi.org/10.1002/nme.4302>.

[37] O.P. Bruno, E. Garza, A Chebyshev-based rectangular-polar integral solver for scattering by general geometries described by non-overlapping patches, 2018, arXiv preprint [arXiv:1807.01813](https://arxiv.org/abs/1807.01813).

[38] C.S. Peskin, J. Comput. Phys. (ISSN: 10902716) 10 (2) (1972) 252–271, [http://dx.doi.org/10.1016/0021-9991\(72\)90065-4](http://dx.doi.org/10.1016/0021-9991(72)90065-4).

[39] G. Iaccarino, R. Verzicco, Appl. Mech. Rev. (ISSN: 00036900) 56 (3) (2003) 331–347, <http://dx.doi.org/10.1115/1.1563627>.

[40] A. Wirth, Ocean Model. (ISSN: 14635003) 9 (1) (2005) 71–87, <http://dx.doi.org/10.1016/j.ocemod.2004.04.003>.

[41] D. Kolomenskiy, K. Schneider, J. Comput. Phys. (ISSN: 0021-9991) 228 (16) (2009) 5687–5709, <http://dx.doi.org/10.1016/j.jcp.2009.04.026>, URL <http://www.sciencedirect.com/science/article/pii/S0021999109002149>.

[42] C. Canuto, A. Quarteroni, M.Y. Hussaini, T.A. Zang, Spectral Methods – Evolution to Complex Geometries and Applications to Fluid Dynamics, in: Scientific Computation, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, <http://link.springer.com/10.1007/978-3-540-30728-0>.

[43] N. Albin, O.P. Bruno, J. Comput. Phys. (ISSN: 00219991) 230 (16) (2011) 6248–6270, <http://dx.doi.org/10.1016/j.jcp.2011.04.023>.

[44] O.P. Bruno, M. Cubillos, J. Comput. Phys. (ISSN: 10902716) 307 (February) (2016) 476–495, <http://dx.doi.org/10.1016/j.jcp.2015.12.010>.

[45] O.P. Bruno, M. Lyon, J. Comput. Phys. (ISSN: 00219991) 229 (6) (2010) 2009–2033, <http://dx.doi.org/10.1016/j.jcp.2009.11.020>.

[46] M. Lyon, SIAM J. Sci. Comput. (ISSN: 1064-8275) 33 (6) (2011) 3241–3260, <http://dx.doi.org/10.1137/11082436X>.

[47] S. Orszag, M. Israeli, Annu. Rev. Fluid Mech. (ISSN: 0066-4189) 6 (1) (1974) 281–318, <http://dx.doi.org/10.1146/annurev.fl.06.010174.001433>.

[48] D. Rempfer, Appl. Mech. Rev. (ISSN: 00036900) 59 (3) (2006) 107, <http://dx.doi.org/10.1115/1.2177683>.

[49] B.T. Kress, D.C. Montgomery, J. Plasma Phys. (ISSN: 00223778) 64 (4) (2000) 371–377, <http://dx.doi.org/10.1017/S0022377800008825>.

[50] G. Gallavotti, Foundations of Fluid Dynamics, Springer-Verlag, 2002.

[51] S.A. Orszag, M. Israeli, M.O. Deville, J. Sci. Comput. (ISSN: 0885-7474) 1 (1) (1986) 75–111, <http://dx.doi.org/10.1007/BF01061454>.

[52] M. Bocher, Ann. of Math. (ISSN: 0003486X) 7 (3) (1906) 81–152, <http://www.jstor.org/stable/1967238>.

[53] N. Albin, O.P. Bruno, T.Y. Cheung, R.O. Cleveland, J. Acoust. Soc. Am. (ISSN: 00014966) 132 (4) (2012) 2371, <http://dx.doi.org/10.1121/1.4742722>.

[54] O.P. Bruno, A. Prieto, J. Sci. Comput. (ISSN: 08857474) 58 (2) (2014) 331–366, <http://dx.doi.org/10.1007/s10915-013-9734-8>.

[55] F. Amlani, O.P. Bruno, J. Comput. Phys. (ISSN: 10902716) 307 (2016) 333–354, <http://dx.doi.org/10.1016/j.jcp.2015.11.060>.

[56] O.P. Bruno, M. Cubillos, E. Jimenez, J. Comput. Phys. (ISSN: 10902716) 391 (2019) 322–346, <http://dx.doi.org/10.1016/j.jcp.2019.02.033>.

[57] A.J. Chorin, Math. Comp. (ISSN: 0025-5718) 22 (104) (1968) 745–745, <http://dx.doi.org/10.1090/S0025-5718-1968-0242392-2>.

[58] J. Kim, P. Moin, J. Comput. Phys. (ISSN: 00219991) 59 (2) (1985) 308–323, [http://dx.doi.org/10.1016/0021-9991\(85\)90148-2](http://dx.doi.org/10.1016/0021-9991(85)90148-2).

[59] J.W. Cooley, J.W. Tukey, Math. Comp. (ISSN: 00255718) 19 (90) (1965) 297, <http://dx.doi.org/10.2307/2003354>.

[60] S.A. Orszag, J. Atmos. Sci. 28 (6) (1971) 1074–1074, [http://dx.doi.org/10.1175/1520-0469\(1971\)028<1074:OTEAOI>2.0.CO;2](http://dx.doi.org/10.1175/1520-0469(1971)028<1074:OTEAOI>2.0.CO;2).

[61] P.D. Mininni, D. Rosenberg, R. Reddy, A. Pouquet, Parallel Comput. (ISSN: 01678191) 37 (6–7) (2011) 316–326, <http://dx.doi.org/10.1016/j.parco.2011.05.004>.

[62] M. Frigo, S.G. Johnson, Proc. IEEE (ISSN: 00189219) 93 (2) (2005) 216–231, <http://dx.doi.org/10.1109/JPROC.2004.840301>.

[63] Nvidia Corporation, CUFFT Library user's guide, 2019, [https://docs.nvidia.com/cuda/pdf/CUFFT\\_Library.pdf](https://docs.nvidia.com/cuda/pdf/CUFFT_Library.pdf). (Accessed 21 January 2020).

[64] R.D. Moser, J. Kim, N.N. Mansour, Phys. Fluids (ISSN: 10706631) 11 (4) (1999) 943–945, <http://dx.doi.org/10.1063/1.869966>.

[65] S. Hoyas, J. Jiménez, Phys. Fluids (ISSN: 10706631) 18 (1) (2006) 10–14, <http://dx.doi.org/10.1063/1.2162185>.

[66] M. Bernardini, S. Pirozzoli, P. Orlandi, J. Fluid Mech. (ISSN: 14697645) 742 (2014) 171–191, <http://dx.doi.org/10.1017/jfm.2013.674>.

[67] A. Lozano-Durán, J. Jiménez, Phys. Fluids (ISSN: 10897666) 26 (1) (2014) 8–15, <http://dx.doi.org/10.1063/1.4862918>.

[68] H. Eckelmann, J. Fluid Mech. (ISSN: 14697645) 65 (3) (1974) 439–459, <http://dx.doi.org/10.1017/S0022112074001479>.

[69] G. Lemout, J.-L. Aider, J.E. Wesfreid, Phys. Rev. E 85 (2) (2012) 025303.

[70] M.P. Schultz, K.A. Flack, International Symposium on Turbulence and Shear Flow Phenomena, TSFP 2013, vol. 2, p. 0–13, <http://dx.doi.org/10.1063/1.4791606>.

[71] M. Lee, R.D. Moser, 2015, <https://turbulence.oden.utexas.edu/>. [Last accessed 15<sup>th</sup> November 2019].

[72] J. Clyne, P. Mininni, A. Norton, M. Rast, New J. Phys. 9 (8) (2007) 301–301.

[73] A. Belmonte, A. Tilgner, A. Libchaber, Phys. Rev. E (ISSN: 1063651X) 50 (1) (1994) 269–279, <http://dx.doi.org/10.1103/PhysRevE.50.269>.

[74] P.A. Davidson, Turbulence in Rotating, Stratified and Electrically Conducting Fluids, Cambridge University Press, 2013.

[75] J. Sakakibara, R.J. Adrian, Exp. Fluids (ISSN: 07234864) 37 (3) (2004) 331–340, <http://dx.doi.org/10.1007/s00348-004-0821-3>.

[76] J.J. Niemela, L. Skrbek, K.R. Sreenivasan, R.J. Donnelly, Nature (ISSN: 00280836) 406 (6794) (2000) 439, <http://dx.doi.org/10.1038/35019124>.

[77] G. Grötzbach, J. Comput. Phys. (ISSN: 10902716) 49 (2) (1983) 241–264, [http://dx.doi.org/10.1016/0021-9991\(83\)90125-0](http://dx.doi.org/10.1016/0021-9991(83)90125-0).

[78] R.M. Clever, F.H. Busse, J. Fluid Mech. (ISSN: 00221120) 344 (1997) 137–153, <http://dx.doi.org/10.1017/S0022112097005818>, [http://www.journals.cambridge.org/abstract\\_...S0022112097005818](http://www.journals.cambridge.org/abstract_...S0022112097005818).

[79] S. Chandrasekhar, Hydrodynamic and Hydromagnetic Stability, Dover Edit, Dover Publication, ISBN: 978-0486640716, 1981.