mathematics

MDPI

Article

# A Code for Simulating Heat Transfer in Turbulent Channel Flow

**Federico Lluesma-Rodríguez, Francisco Álcantara-Ávila, María Jezabel Pérez-Quiles and Sergio Hoyas ***

Instituto Universitario de Matemática Pura y Aplicada, Universitat Politècnica de València, 46022 València, Spain; fellrod@doctor.upv.es (F.L.-R.); fraalvi@alumni.upv.es (F.Á.-Á.); jperezq@mat.upv.es (M.J.P.-Q.)
* Correspondence: serhocal@mot.upv.es

**Abstract:** One numerical method was designed to solve the time-dependent, three-dimensional, incompressible Navier–Stokes equations in turbulent thermal channel flows. Its originality lies in the use of several well-known methods to discretize the problem and its parallel nature. Vorticy-Laplacian of velocity formulation has been used, so pressure has been removed from the system. Heat is modeled as a passive scalar. Any other quantity modeled as passive scalar can be very easily studied, including several of them at the same time. These methods have been successfully used for extensive direct numerical simulations of passive thermal flow for several boundary conditions.

**Keywords:** DNS; CFD; turbulence

check for updates

**Citation:** Lluesma-Rodríguez, F.; Álcantara-Ávila, F.; Pérez-Quiles, M.J.; Hoyas, S. A Code for Simulating Heat Transfer in Turbulent Channel Flow. *Mathematics* **2021**, *9*, 756. https://doi.org/10.3390/math9070756

Academic Editor: Yolanda Vidal

Received: 28 February 2021
Accepted: 30 March 2021
Published: 1 April 2021

## 1. Introduction

Turbulence is probably the open problem in physics with most applications in daily life. Turbulent flows are intrinsic to almost any flow in engineering, but they are also extremely important in meteorology or the dispersion of contaminants. It is well known that we still even lack an existence and uniqueness theorem about the solution of the governing equations of turbulent flows, the Navier–Stokes equations. Moreover, these equations cannot be solved analytically apart from some trivial examples. It is generally admitted that the simulation can be considered at three levels of detail, Reynolds Average Navier Stokes (RANS), Large Eddy Simulation (LES), and Direct Numerical Simulations (DNS) [1] which also corresponds to different levels of precision. Both RANS and LES models can simulate typical industrial flows with the necessary accuracy and in practical computation times [2–5], but DNS is until now the only trustworthy method to investigate physical properties of flows which are little understood.

However, DNS are extremely expensive as every scale of turbulence, both temporal and spatial ones, has to be properly resolved. This implies very fine grids and small temporal steps. Moreover, and as a consequence of Kolmogorov's scales, [6,7], the number of points needed to carry out the simulation grows extremely fast, as $Re^{9/4}$, where $Re$ is the Reynolds number. Thus, to use DNS to understand turbulent flows, very simple geometries are employed, allowing very precise and fast numerical tools that will be described in this paper. Canonical wall-bounded flows are pipes, channel flows, and the flat-plate boundary layer. These flows are in essence theoretical abstractions and do not appear as such in reality. However, they constitute the basic blocks of more complete real flows and have been studied for 80 years. For isothermal flows, experiments of these canonical flows have been instrumental in the development of the theory since the birth of boundary layer research. In particular, they remain important for the study of very high Reynolds numbers, as DNS is still not able to reach Reynolds numbers as high as those of experiments. High Reynolds numbers are needed to obtain a clear separation of scales related to the near-wall turbulence cycle. The main control parameter is the friction Reynolds Number $Re_\tau = hu_\tau/\nu$, where $\nu$ is the viscosity, $h$ is a characteristic length and $u_\tau$ is the friction velocity at the wall. The largest experiment ever done has been published in 2018 for a friction Reynolds number of 2e4 [8], a value that is absolutely out of reach for DNS. That said, simulations of wall-bounded turbulence are particularly helpful in identifying physical processes occurring in near-wall turbulence as the whole velocity field (and its derivatives) is available. Moreover, experiments showing the 3D thermal field of a turbulent flow are almost impossible to do. For thermal flows, DNS can give insight where experimental flows are impossible.

The first wall-bounded simulation was run by Kim, Moin, and Moser in 1987 [9] for channel flows and in 1988 by Spalart [10] for Turbulent Boundary Layers. Since then, the friction Reynolds number simulated has been continuously growing [11–19]. Some of these simulations, Ref. [12,16,18,19] were made with an earlier version of this code. The history is similar for thermal passive flows, but the Reynolds number reached grew up at a slower pace. The first DNS of a thermal flow was carried out by Kim and Moin in 1987 [20]. Kim and Moin obtained first-order turbulence statistics for $Re_\tau = 180$ and several Prandtl numbers, $Pr = 0.1$, 1 and 2. The Prandtl number is the ratio between momentum diffusivity to thermal diffusivity. They also computed a very important number for modeling, the turbulent Prandtl number. In addition, for the Prandtl number of air, $Pr = 0.71$, correlations between the velocity and the temperature were also calculated. A somewhat artificial boundary condition was imposed in which heat was generated internally and removed from both cold isothermal walls. This condition for the thermal field plays an analogous role to that of the pressure gradient does for the velocity field. Later, Lyons et al. [21] performed a simulation for $Re_\tau = 150$ and $Pr = 1$. The boundary condition used in this later work consisted of both walls kept at different temperatures. Finally, Kasagi et al. [22] performed a DNS for $Re_\tau = 150$ and $Pr = 0.71$ with a more realistic boundary condition, the Mixed Boundary Condition (MBC from now on). For this condition, the average heat flux over both heating walls is constant and the temperature increases linearly in the streamwise direction. The instantaneous heat flux may vary with respect to time and position. In the work done by Piller [23], it was shown that the MBC acts as an ideal isothermal boundary condition in the inner layer and as an ideal isoflux boundary condition (fixed heat flux) in the outer layer.

After these simulations were made, the trend has been to increase the friction Reynolds number for different molecular Prandtl numbers. However, values of $Re_\tau$ and $Pr$ are limited by the computational cost. Yano and Kasagi [24] stated that the computational cost can be approximated as $L_x^2 L_y Re_\tau^4 Pr^{3/2}$. Moreover, even if only one equation has to be added to the system, this equation is also nonlinear. Typically, the cost of accurately computing the nonlinear term accounts for 80 to 90% of the total time of the simulation. Thus, to add the energy equation we need to double the computational time, and the Reynolds number achieved is still low for the majority of Prandtl numbers.

In this paper, we will restrict ourselves to explain the numerical code and its implementation. Since 2018, several works have been published using this algorithm, which was written in Fortran90. In Lluesma et al. [25], the authors validated the code and found the optimal size of the computational box for Reynolds numbers up to $Re_\tau = 2000$. The range of Prandtl numbers, below $Pr = 0.7$, was increased in Alcántara-Ávila [26]. Later, in [27], the large structures of Couette flow were analyzed. In [28,29], the authors were able to increase the study for $Pr > 0.7$, reaching $Pr = 10$, and $Re_\tau = 5000$ for $Pr = 0.7$. Finally, this code has been used to study the large structures found in stratification problems, cite [30].

The structure of the paper is as follows. In the next section, the numerical problem is described. The third section is devoted to explain the numerical techniques used for discretization both in time and space. The fourth section describes the parallelization techniques used. Finally, conclusions and future works are outlined.

## 2. Methods

The flow considered in this work is a turbulent channel flow driven by a pressure gradient (Poiseuille flow). The flow is treated as incompressible. The thermal field is treated as a passive scalar. As mentioned before, the boundary condition used for the thermal field is the MBC. For this boundary condition, an uniform heat flux is heating both walls, which introduces the heat into the flow. The temperature of these walls increases linearly in the streamwise direction and does not depend on time.

In Figure 1, a schematic representation of the lower half of the computational box used can be observed. In this plot, contours of an instantaneous snapshot of the streamwise velocity are represented, colored by the magnitude of the velocity. The flow moves from left to right. Periodic conditions are imposed in the streamwise and spanwise boundaries.

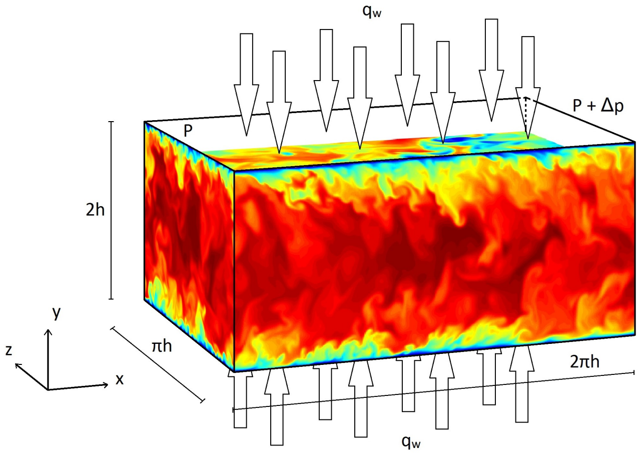

**Figure 1.** Geometry and instantaneous representation of the temperature field.

As it was said above, the computational box was optimized in [25]. The dimensions of the box are $L_x = 2\pi h$, $L_y = 2h$, and $L_z = \pi h$ in the streamwise, wall-normal, and spanwise directions, respectively. Coordinates in these directions are denoted by $x$, $y$, and $z$. The corresponding velocities are $U$, $V$, and $W$, or, using index notation, $U_i$. This is also the case for vorticity, $\vec{\Omega} = \vec{\nabla} \times \vec{U} = (\Omega_1, \Omega_2, \Omega_3) = \left(\Omega_x, \Omega_y, \Omega_z\right)$ and helicity $\vec{H} = \vec{U} \times \vec{\Omega} = (H_1, H_2, H_3) = \left(H_x, H_y, H_z\right)$. Temperature is represented by $T$. However, the transformed temperature, $\Theta$ (defined below), will be used during the entire paper. Uppercase letters denote instantaneous flow magnitudes. Using the Reynolds decomposition, one can obtain the averaged value, denoted by an overbar, and the fluctuating part, denoted by a lowercase letter, of the flow magnitudes, i.e., $U = \overline{U} + u$. The superscript $+$ indicates normalization in wall units, using $\nu$ and $u_\tau = \sqrt{\tau_w/\rho}$, where $\tau_w$ is the mean shear stress and $\rho$ is the fluid density.

The behavior of turbulent flows is described by the Navier–Stokes equations, which are composed by the continuity and momentum equations,

$$\partial_j U_j^+ \;=\; 0, \tag{1}$$

$$\partial_t U_i^+ + U_j^+ \partial_j U_i^+ \;=\; -\partial_i P^+ + \frac{1}{Re_\tau} \partial_{jj} U_i^+, \tag{2}$$

and the energy equation,

$$\partial_t \Theta^+ + U_j^+ \partial_j \Theta^+ = \frac{1}{Re_\tau Pr} \partial_{jj} \Theta^+ + U^+ / \langle U^+ \rangle_{xyz}, \tag{3}$$

where $\langle U^+ \rangle_{xyz}$ is the average of $U^+$ in time and in the three spatial directions, i.e., $\langle U^+ \rangle_{xyz} = U_b^+$, where $U_b^+$ is the bulk velocity. In the energy equation, the transformed temperature, $\Theta = T_w - T$ is used. Since the MBC is used, the temperature in the channel increases linearly in the streamwise direction and periodic conditions cannot be used for $T$. $T_w$ contains the nonperiodic part of the temperature, which makes $\Theta$ periodic in the streamwise direction. Notice that the method described below is also useful for any other passive scalar, as it could be NOx concentration. It is also important to point out that for clarity, only the system with one energy equation will be explained here. However, as heat transfer is treated as a passive scalar, several different Prandtl numbers can be run simultaneously. In what follows and to facilitate the discussion, + superscripts will be omitted.

Using some algebra, which involves taking the rotational of Equations (1) and (2) twice, these equations can be written in velocity-vorticity form obtaining a fourth-order equation for $V$

$$\partial_t \nabla^2 V = -h_v + \frac{1}{Re_\tau} \nabla^4 V, \tag{4}$$

and a second-order equation for $\Omega_y$

$$\partial_t \Omega_y = -h_g + \frac{1}{Re_\tau} \nabla^2 \Omega_y, \tag{5}$$

where $h_v$ and $h_g$ collect the nonlinear part of the Equations (4) and (5), and are given by

$$\begin{aligned} h_v &= -\partial_y(\partial_x H_1 + \partial_z H_3) + (\partial_{xx} + \partial_{zz}) H_2, \\ h_g &= \partial_z H_1 - \partial_x H_3, \end{aligned}$$

The main advantage of this system is that pressure is not present, simplifying the problem. Notice that one can easily recover velocities and vorticities from the continuity equation and the definition of $\Omega_y$,

$$\partial_x U + \partial_z W = -\partial_y V, \tag{6}$$

$$\partial_z U - \partial_x W = \Omega_y, \tag{7}$$

Taking derivatives,

$$\partial_x(\partial_x U + \partial_z W) + \partial_z(\partial_z U - \partial_x W) = \partial_x(-\partial_y V) + \partial_z(\Omega_y), \tag{8}$$

$$\partial_z(\partial_x U + \partial_z W) - \partial_x(\partial_z U - \partial_x W) = \partial_z(-\partial_y V) + \partial_x(\Omega_y), \tag{9}$$

and reorganizing the terms, we get

$$(\partial_{xx} + \partial_{zz}) U = -\partial_{yx} V + \partial_z \Omega_y, \tag{10}$$

$$(\partial_{xx} + \partial_{zz}) W = -\partial_{yz} V - \partial_x \Omega_y. \tag{11}$$

As the flow is periodic in both $x$ and $z$, it is natural to use Fourier methods in these directions. The Fourier transform of any field $\varphi(x, y, z)$ is defined as

$$\varphi(x, y, z) = \sum_{k_x} \sum_{k_z} \hat{\varphi}_{k_x k_z}(y) e^{i(k_x x + k_z z)},$$

where the hat indicates the Fourier coefficient, and $k_x$ and $k_z$ are the wavenumbers in $x$ and $z$, respectively. Equations (10) and (11) can be trivially solved as they become

$$\hat{U}_{k_x k_z} = \frac{-ik_x \partial_y \hat{V}_{k_x k_z} + i_{k_z} \hat{\Omega}_y}{-(k_x^2 + k_z^2)}, \tag{12}$$

$$\hat{W}_{k_x k_z} = \frac{-i_{k_z} \partial_y \hat{V}_{k_x k_z} - ik_x \hat{\Omega}_{y_{k_x k_z}}}{-(k_x^2 + k_z^2)}. \tag{13}$$

Notice that this technique cannot be used for the $(0, 0)$-modes of $U$ and $W$. Using the Navier–Stokes equations, one can write

$$\begin{aligned} \partial_t \hat{U}_{00} &= -\partial_x \hat{P} + \hat{H}_{1,00} + \frac{1}{Re_\tau} \partial_{yy} \hat{U}_{00}, \\ \partial_t \hat{W}_{00} &= -\partial_z \hat{P} + \hat{H}_{3,00} + \frac{1}{Re_\tau} \partial_{yy} \hat{W}_{00}, \end{aligned}$$

and solve these equations instead. The pressure gradient in the spanwise direction is negligible. However, the pressure gradient in the streamwise component can be used to keep the flow mass constant. Using Fourier transforms, Equations (3)–(5) are transformed into Fourier space in $x$ and $z$, obtaining three decoupled problems,

$$\begin{aligned}
\partial_t \hat{\phi}_{k_x k_z} &= \hat{h}_{v,k_x k_z} + \frac{1}{Re_\tau} \nabla^2 \hat{\phi}_{k_x k_z}, && (14)\\
\nabla^2 \hat{V}_{k_x k_z} &= \hat{\phi}_{k_x k_z}, && (15)\\
\hat{V}_{k_x k_z}(\pm 1) &= 0,\ \partial_y \hat{V}_{k_x k_z}(\pm 1) = 0, && (16)\\
\partial_t \hat{\Omega}_{y,k_x k_z} &= \hat{h}_{g,k_x k_z} + \frac{1}{Re_\tau} \nabla^2 \hat{\Omega}_{y,k_x k_z}, && (17)\\
\hat{\Omega}_{y k_x k_z}(\pm 1) &= 0, && (18)\\
\partial_t \hat{\Theta}_{k_x k_z} &= \hat{h}_{t,k_x k_z} + \frac{1}{Re_\tau Pr} \nabla^2 \hat{\Theta}_{k_x k_z}, && (19)\\
\hat{\Theta}_{k_x k_z}(\pm 1) &= 0, && (20)
\end{aligned}$$

where $\hat{h}_{t,k_x k_z}$ is the Fourier transform of the energy equation nonlinear term.

Problem (14)–(16) is usually solved splitting $\hat{V}_{k_x k_z}$ in three parts, (wavenumbers are omitted)

$$\hat{V} = \hat{V}_p + a\hat{V}_a + b\hat{V}_b,$$

where $a$ and $b$ are chosen in order to fulfill the homogeneous Neumann condition:

$$\begin{cases} \partial_t \hat{\phi}_p = \hat{h}_v - Re_\tau^{-1}(k_x^2 + k_z^2 - \partial_{yy})\hat{\phi}_p, \\ \hat{\phi}_p(\pm 1) = 0, \\ -(k_x^2 + k_z^2 - \partial_{yy})\hat{V}_p = \hat{\phi}_p, \\ \hat{V}_p(\pm 1) = 0, \end{cases} \tag{21}$$

$$\begin{cases} \partial_t \hat{\phi}_a = -Re_\tau^{-1}(k_x^2 + k_z^2 - \partial_{yy})\hat{\phi}_a, \\ \hat{\phi}_a(\pm 1) = 0, \\ -(k_x^2 + k_z^2 - \partial_{yy})\hat{V}_a = \hat{\phi}_a, \\ \hat{V}_a(1) = 1, \hat{V}_a(-1) = 0, \end{cases} \tag{22}$$

$$\begin{cases} \partial_t \hat{\phi}_b = -Re_\tau^{-1}(k_x^2 + k_z^2 - \partial_{yy})\hat{\phi}_b, \\ \hat{\phi}_b(\pm 1) = 0, \\ -(k_x^2 + k_z^2 - \partial_{yy})\hat{V}_b = \hat{\phi}_b, \\ \hat{V}_b(1) = 0, \hat{V}_b(-1) = 1, \end{cases} \tag{23}$$

and

$$\begin{cases} \partial_t \hat{\Omega}_{y,kxkz} = \hat{h}_g - Re_\tau^{-1}(k_x^2 + k_z^2 - \partial_{yy})\hat{\Omega}_{y,kxkz}, \\ \hat{\Omega}_y(\pm 1) = 0, \end{cases} \tag{24}$$

$$\begin{cases} \partial_t \hat{\Theta}_{kxkz} = \hat{h}_t - (Re_\tau Pr)^{-1}(k_x^2 + k_z^2 - \partial_{yy})\hat{\Theta}_{kxkz}, \\ \hat{\Theta}_{kxkz}(\pm 1) = 0, \end{cases} \tag{25}$$

although, due to the symmetry of the problem, $\hat{V}_b(y) = \hat{V}_a(-y)$, so only Equations (21), (22), (24) and (25) have to be solved. Notice that 3D PDE have been transformed into $N_x \times N_z$ independent problems.

## 3. Numerical Method

### *3.1. CFD Techniques*

As it is said above, Fourier method in $x$ and $z$ is the best option to solve this problem. This, in fact, decouples the problems (21)–(25) in a large amount of problems in $y$. The numerical technique used for $y$ is thus critical to obtain an accurate and fast algorithm. This kind of problem had been addressed mostly by spectral methods [9,31]. A typical technique used in channels is Chebyshev polynomials [11,32]. This has also been applied to thermoconvective problems [33–35]. However, in this case, it is more efficient the use of Compact Finite Differences, CFD. The CFD method was introduced in a groundbreaking article by SK Lele in 1992 [36]. Lele's idea was to use finite differences to solve problems presenting a range of spatial scales, generalizing some Padé schemes that had been used early [37,38]. Lele's CFD main advantage is that maintaining the freedom in choosing the mesh points of typical finite difference methods (FD), offering very high precision, comparable to spectral methods. This is critical in turbulent problems, as one typically needs many more points close to the wall than in the outer regions of the flow.

Throughout this part of this work, the following notation is going to be used. Let $u(y)$ be a real evaluated function. Supposing that $u$ is differentiable enough, the first and second derivative of $u$ at the point $y_i$ will be denoted by $u'_i = u'(y_i)$ and $u''_i = u''(y_i)$. In the case of a derivative of grade $n$, we will use $u_i^{n)} = \frac{d^n}{dy^n}u(y_i)$. Here, $y_i$ is a point belonging to a certain discretization of the interval $[a, b]$, where $a$ and $b$ are finite and $y_0 = a$, $y_n = b$. Without loss of generality, we will focus on schemes for the second derivative. FD schemes aim to compute an approximation of $u''_i$ trough a linear combination of the values of the function close to $y_i$. CFD, instead, relates a linear combination of the second derivatives with the values of the function. To explain the practical use of CFD schemes, we are going to work with one specific example, the scheme used in [29]. In that work, the authors used a stencil of seven points in the function and five in the second derivative. Let us assume that the following relation holds

$$\alpha_{i-2}u''_{i-2} + \alpha_{i-1}u''_{i-1} + \alpha_0 u''_i + \alpha_{i+1}u''_{i+1} + \alpha_{i+2}u''_{i+2} \tag{26}$$

$$= a_{i-3}u_{i-3} + a_{i-2}u_{i-2} + a_{i-1}u_{i-1} + a_0 u_i + a_{i+1}u_{i+1} + a_{i+2}u_{i+2} + a_{i+3}u_{i+3}, \tag{27}$$

for some unknowns coefficients, $\alpha_j$ and $a_j$. Without loss of generality, it is possible to assume that one of these coefficients is one, so from now on, $\alpha_0 = 1$. Defining $h_j = (y_i - y_{i+j})$, Taylor's Theorem states that

$$\begin{aligned}
u_{i-3} &= u_i + \frac{h_{-3}}{1!}u'_i + \frac{h_{-3}^2}{2!}u''_i + \cdots + \frac{h_{-3}^n}{n!}u_i^{n)} + O\left(h_{-3}^{n+1}\right),\\
u_{i-2} &= u_i + \frac{h_{-2}}{1!}u'_i + \frac{h_{-2}^2}{2!}u''_i + \cdots + \frac{h_{-2}^n}{n!}u_i^{n)} + O\left(h_{-2}^{n+1}\right),\\
u_{i-1} &= u_i + \frac{h_{-1}}{1!}u'_i + \frac{h_{-1}^2}{2!}u''_i + \cdots + \frac{h_{-1}^n}{n!}u_i^{n)} + O\left(h_{-1}^{n+1}\right),\\
u_i &= u_i,\\
u_{i+1} &= u_i + \frac{h_1}{1!}u'_i + \frac{h_1^2}{2!}u''_i + \cdots + \frac{h_1^n}{n!}u_i^{n)} + O\left(h_1^{n+1}\right),\\
u_{i+2} &= u_i + \frac{h_2}{1!}u'_i + \frac{h_2^2}{2!}u''_i + \cdots + \frac{h_2^n}{n!}u_i^{n)} + O\left(h_2^{n+1}\right),\\
u_{i+3} &= u_i + \frac{h_3}{1!}u'_i + \frac{h_3^2}{2!}u''_i + \cdots + \frac{h_3^n}{n!}u_i^{n)} + O\left(h_3^{n+1}\right),
\end{aligned}$$

and

$$\begin{aligned}
u''_{i-2} &= u''_i + \frac{h_{-2}}{1!}u_i^{3)} + \frac{h_{-2}^2}{2!}u_i^{4)} + \cdots + \frac{h_{-2}^n}{n!}u_i^{n+2)} + O\left(h_{-2}^{n+1}\right),\\
u''_{i-1} &= u''_i + \frac{h_{-1}}{1!}u_i^{3)} + \frac{h_{-1}^2}{2!}u_i^{4)} + \cdots + \frac{h_{-1}^n}{n!}u_i^{n+2)} + O\left(h_{-1}^{n+1}\right),\\
u''_i &= u''_i,\\
u''_{i+1} &= u''_i + \frac{h_1}{1!}u_i^{3)} + \frac{h_1^2}{2!}u_i^{4)} + \cdots + \frac{h_1^n}{n!}u_i^{n+2)} + O\left(h_1^{n+1}\right),\\
u''_{i+2} &= u''_i + \frac{h_2}{1!}u_i^{3)} + \frac{h_2^2}{2!}u_i^{4)} + \cdots + \frac{h_2^n}{n!}u_i^{n+2)} + O\left(h_2^{n+1}\right).
\end{aligned}$$

The relations between the coefficients $a_i$ and $\alpha_i$ are derived by matching the Taylor series coefficients. In this case, the formal truncation error of the approximation is tenth order. Translating this information into an algebraic equation leads to the system

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & h_{-3} & h_{-2} & h_{-1} & 0 & h_1 & h_2 & h_3 \\ -1 & -1 & -1 & -1 & \frac{h^2_{-3}}{2!} & \frac{h^2_{-2}}{2!} & \frac{1h^2_{-1}}{2!} & 0 & \frac{h_1^2}{2!} & \frac{h_2^2}{2!} & \frac{h_2^2}{2!} \\ -h_{-2} & -h_{-1} & -h_1 & -h_2 & \frac{h^3_{-3}}{3!} & \frac{h^3_{-2}}{3!} & \frac{h^3_{-1}}{3!} & 0 & \frac{h_1^3}{3!} & \frac{h_2^3}{3!} & \frac{h_3^3}{3!} \\ -h^2_{-2} & -h^2_{-1} & -h_1^2 & -h_2^2 & \frac{h^4_{-3}}{4\cdot 3} & \frac{h^4_{-2}}{4\times 3} & \frac{h^4_{-1}}{4\times 3} & 0 & \frac{h_1^4}{4\times 3} & \frac{h_2^4}{4\times 3} & \frac{h_3^4}{4\times 3} \\ -h^3_{-2} & -h^3_{-1} & -h_1^3 & -h_2^3 & \frac{h^5_{-3}}{5\times 4} & \frac{h^5_{-2}}{5\times 4} & \frac{h^5_{-1}}{5\times 4} & 0 & \frac{h_1^5}{5\times 4} & \frac{h_2^5}{5\times 4} & \frac{h_3^5}{5\times 4} \\ -h^4_{-2} & -h^4_{-1} & -h_1^4 & -h_2^4 & \frac{h^6_{-3}}{6\times 5} & \frac{h^6_{-2}}{6\times 5} & \frac{h^6_{-1}}{6\times 5} & 0 & \frac{h_1^6}{6\times 5} & \frac{h_2^6}{6\times 5} & \frac{h_3^6}{6\times 5} \\ -h^5_{-2} & -h^5_{-1} & -h_1^5 & -h_2^5 & \frac{h^7_{-3}}{7\times 6} & \frac{h^7_{-2}}{7\times 6} & \frac{h^7_{-1}}{7\times 6} & 0 & \frac{h_1^7}{7\times 6} & \frac{h_2^7}{7\times 6} & \frac{h_3^7}{7\times 6} \\ -h^6_{-2} & -h^6_{-1} & -h_1^6 & -h_2^6 & \frac{h^8_{-3}}{8\times 7} & \frac{h^8_{-2}}{8\times 7} & \frac{h^8_{-1}}{8\times 7} & 0 & \frac{h_1^8}{8\times 7} & \frac{h_2^8}{8\times 7} & \frac{h_3^8}{8\times 7} \\ -h^7_{-2} & -h^7_{-1} & -h_1^7 & -h_2^7 & \frac{h^9_{-3}}{9\times 8} & \frac{h^9_{-2}}{9\times 8} & \frac{h^9_{-1}}{9\times 8} & 0 & \frac{h_1^9}{9\times 8} & \frac{h_2^9}{9\times 8} & \frac{h_3^9}{9\times 8} \\ -h^8_{-2} & -h^8_{-1} & -h_1^8 & -h_2^8 & \frac{h^{10}_{-3}}{10\times 9} & \frac{h^{10}_{-2}}{10\times 9} & \frac{h^{10}_{-1}}{10\times 9} & 0 & \frac{h_1^{10}}{10\times 9} & \frac{h_2^{10}}{10\times 9} & \frac{h_3^{10}}{10\times 9} \end{pmatrix} \begin{pmatrix} \alpha_{i-2} \\ \alpha_{i-1} \\ \alpha_{i+1} \\ \alpha_{i+2} \\ a_{i-3} \\ a_{i-2} \\ a_{i-1} \\ a_i \\ a_{i+1} \\ a_{i+2} \\ a_{i+3} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Please note that this is the direct matrix coming from matching Taylor's expansions. Usually, this matrix is ill-conditioned, so results can be very inaccurate. A typical procedure to overcome this problem consists of normalizing each row by its absolute value maximum.

As the mesh can be nonuniform, it is necessary to solve this system for each point in the mesh. When approaching the boundaries, no ghost points are used but the stencils are adapted, removing the points which lay outside the interval. This reduces the formal truncation error of the system. Once the coefficients for every point have been computed, we obtain two sparse matrices, one containing the $a_i$ coefficients, $A_{yy}$, and another one made by the $\alpha_i$ coefficients, $B_{yy}$, such that

$$B_{yy}(u_0'', ..., u_n'')^t = A_{yy}(u_0, ..., u_n)^t, \tag{28}$$

where the superscript $()^t$ denotes the transpose. Note that (28) can be used in both ways, to derive a function or to integrate it.

### 3.2. Time Discretization

To get good accuracy in reasonable computation times, a third-order Runge–Kutta method has been chosen, derived in [39]. For any problem with the form

$$\partial_t \varphi = L(\varphi) + N(\varphi), \tag{29}$$

where $L$ is a linear operator and $N$ is nonlinear, the equation is discretized as follows:

$$(I - \beta_1 \Delta t L)\varphi_1 = (I + \Delta t \alpha_1 L)\varphi_0 + \Delta t \gamma_1 N_0, \tag{30}$$

$$(I - \beta_2 \Delta t L)\varphi_2 = (I + \Delta t \alpha_2 L)\varphi_1 + \Delta t(\gamma_2 N_1 + \xi_1 N_0), \tag{31}$$

$$(I - \beta_3 \Delta t L)\varphi_3 = (I + \Delta t \alpha_3 L)\varphi_2 + \Delta t(\gamma_3 N_2 + \xi_2 N_1), \tag{32}$$

with $\varphi_i = \varphi(t^i)$, for $t^n = t^0, t^1, t^2, t^3 = t^{n+1}$. This method presents two problems. The first one is a problem of memory, due to the necessity of saving two nonlinear terms in steps two and three. The second problem is a possible loss of accuracy near the wall, due to the explicit computation of the linear operator in the right hand,

$$L_i^t \varphi_0 = (I + \Delta t \alpha_i L)\varphi_0.$$

Both problems are solved in the algorithm, using a little bit of algebra. Suppose that (30) is solved, and $\varphi_2$ is to be computed. Let us denote

$$Rh_1 = (I + \Delta t \alpha_1 L)\varphi_0 + \Delta t \gamma_1 N_0,$$

then

$$L\varphi_1 = \frac{\varphi_1 - Rh_1}{\beta_1 \Delta t},$$

and we can compute and store in a single buffer of memory the right hand side of (31) except the nonlinear term $N_1$. This tactic can be also applied in the computation of $\varphi_1$ but only in the cases where $\Delta t$ does not change. If the time step changes, the derivative of $\varphi_0$ has to be computed explicitly. To avoid this loss of accuracy, it is a good idea to recompute the maximum $\Delta t$ every few steps, using the Courant–Friedichs–Lewy condition

$$\max\left\{\frac{U\Delta t}{\Delta x}, \frac{V\Delta t}{\Delta y}, \frac{W\Delta t}{\Delta z}\right\} \le CFL.$$

The simulations made with this code ran with a CFL of 0.7, showing remarkable stability.

Equations (30)–(32) are solved for $\phi$, $\Omega_y$ and $\Theta$ using CFD. In what follows we will work with the energy equation, given that the linear operator is $L = Re_\tau^{-1}\nabla^2$ for $\phi$ and $\Omega_y$, and $L = (PrRe_\tau)^{-1}\nabla^2$ for $\Theta$. The operator $L$ in Fourier space can be written as

$$L = (Re_\tau Pr)^{-1}(\partial_{xx} + \partial_{yy} + \partial_{zz}) \Rightarrow \hat{L}_{k_x k_z} = (Re_\tau Pr)^{-1}((ik_x)^2 + (ik_z)^2 + \partial_{yy}).$$

Denoting by $RHS_i$ the right-hand side of (30)–(32), this system becomes

$$(I - \beta_i \Delta t L)\varphi_i = RHS_i. \tag{33}$$

In Fourier space, one gets

$$\left(I - \beta_i \Delta t \frac{-k_x^2 - k_z^2 + \partial_{yy}}{Re_\tau Pr}\right)\varphi_i = RHS_i, \tag{34}$$

$$\left(I + \beta_i \Delta t \frac{k_x^2 + k_z^2}{Re_\tau Pr}\right)\varphi_i - \frac{\beta_i \Delta t}{Re_\tau Pr}\partial_{yy}\varphi_i = RHS_i, \tag{35}$$

Now using the two matrix described in the CFD section (28), and defining $\eta$ as the first constant of the previous equation,

$$\eta = \left(I + \beta_i \Delta t \frac{k_x^2 + k_z^2}{Re_\tau Pr}\right),$$

we get

$$\eta\varphi_i - \frac{\beta_i \Delta t}{Re_\tau Pr}\partial_{yy}\varphi_i = RHS, \tag{36}$$

$$\eta B_{yy}\varphi_i - \frac{\beta_i \Delta t}{Re_\tau Pr}B_{yy}\partial_{yy}\varphi_i = B_{yy}RHS, \tag{37}$$

$$\eta B_{yy}\,\varphi_i - \frac{\beta_i \Delta t}{Re_\tau Pr}A_{yy}\varphi_i = B_{yy}RHS. \tag{38}$$

Defining a new matrix $M$, as

$$M = \eta B_{yy} - \frac{\beta_i \Delta t}{Re_\tau Pr}A_{yy}, \tag{39}$$

the final problem to solve is

$$M\varphi_i = B_{yy}RHS.$$

Notice that this matrix is banded, which allows low-storage schemes and the use of very efficient LAPACK routines.

## 4. Parallelization Strategy

The most expensive part of the algorithm is the computation of the nonlinear terms $\hat{h}_v$, $\hat{h}_g$ and $\hat{h}_t$. This is because it is necessary to compute these terms in physical space, due to the aliasing problem [31]. Notice that there are two global operations:

1. Direct and inverse Fourier transforms in $x$ and $z$, as the nonlinear term has to be computed in physical space.
2. Integration-derivation in $y$.

As the data is distributed throughout the supercomputer, it is necessary to perform several all-to-all communications, which are critical and extremely demanding of the fast network of the supercomputer. The code demands a total of $comm = 3 \times (9 + 3h_p)$ global communications per step, where $h_p$ is the total number of heat transfer problems studied.

The number of points of the problem, and thus memory, depend on the Reynolds number studied. Another constraint is the efficiency of the fast Fourier transforms (FFT), so typically numbers made up of powers of 2 and 3 are chosen to increase the velocity of the FFT. Assuming a mesh with $mx \times my \times mz$ points, generally $mx$ is the largest one. It is then natural to start each substep of the Runge–Kutta scheme with the data set distributed in $y$–$z$ planes. To avoid load imbalances, the number of nodes must be a divisor of the number of complex planes, which is $mx/2$ after dealiasing. This number is also the maximum number of nodes that can be used. The maximum number of OpenMP processes at each node is $mx/2/nprocs$, where $nprocs$ is the number of processors of the node.

The code to implement the algorithm described above, where Figure 2 can serve as roadmap, is as follows

1. Read data and configuration files: HDF5
2. Runge–Kutta, data: $\phi$, $\Omega_y$ and $\Theta$, in Fourier Space, distributed through the supercomputer with a $(y, z, x)$ shape.
   (a) Calculation of nonlinear terms (I). Time: 15% of total time
      i. $\phi, \Omega_y \rightarrow \vec{U}, \vec{\Omega}$.
      ii. Compute statistical quantities of the flow if needed.
      iii. Change shape from $(y, z, x)$ to $(z, y, x)$.
      iv. Perform inverse transforms in $z$ of $\vec{U}$ and $\vec{\Omega}$.
   (b) Calculation on nonlinear terms (II). Time: 35% of total time
      i. Global transposes: Data is moved trough all the machine, from $y$–$z$ planes into $x$-lines using MPI routines.
   (c) Calculation on nonlinear terms (III) . Time:10% of total time
      i. Perform inverse transforms in $x$ of $\vec{U}$ and $\vec{\Omega}$.
      ii. Computes three technical buffers $C_i$ to compute later the nonlinear terms $\hat{h}_v$ and $\hat{h}_g$. $h_t$ is computed here.
      iii. Perform direct transforms in $x$ of $\vec{U}$ and $\vec{\Omega}$.
   (d) Calculation on nonlinear terms (IV). Time: 20% of total time
      i. Global transposes: Data is moved trough all the machine, from $x$-lines into $y$–$z$ planes
   (e) Calculation on nonlinear terms (V). Time: 10% of total time
      i. Perform direct transforms in $z$ of $C_i$ and $h_t$.
      ii. Change shape from $(z, y, x)$ to $(y, z, x)$.
   (f) Solve viscous problem. This step requires 10% of total time.
   (g) Save data to disk if needed.
   (h) Move to the step 2a.

3. End of program.

The code spends 90% of the time and almost all communication time computing the nonlinear term and only 10% to actually solve the equations. A large share of this 90% is used to do global communications. This is a critical step and requires an in-deep study of the supercomputer where the simulation is going to run.

Finally, notice that while global communications uses MPI routines, the granularity of the code makes the use of OpenMP routines for the Fourier transforms, derivatives, and the viscous step. This is an important advantage due to the foreseen evolution of supercomputation.

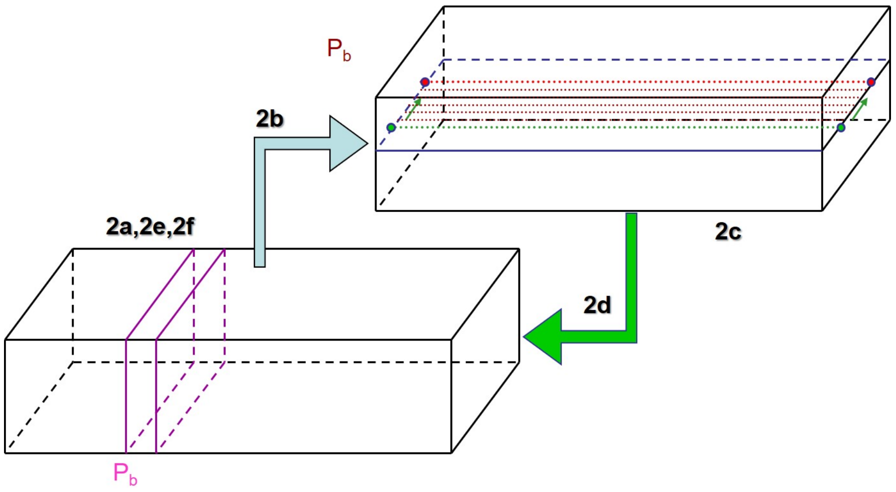

**Figure 2.** Parallelization scheme of the code.

## 5. Conclusions

The main goal of this work is to develop a code to perform direct numerical simulations of distinctive wall-bounded thermal canonical flows. This code has run for roughly 50 CPU-M hours in several supercomputers, already producing several articles. FFT and CFD techniques have been described—in particular, how to use a nonequispaced grid for CFD. In addition, the combination of the Runge–Kutta time-stepper and the previous techniques have been explained. Finally, the parallelization scheme has been outlined. Its main advantage is the granularity of the data, allowing a large amount of small problems in each step that can be solved efficiently using OpenMP techniques.

The raw data that support the findings of this study are available from the corresponding author upon reasonable request. One point statistics can be downloaded from the web page of our group : http://personales.upv.es/serhocal/ (accessed on 24 March 2021).

**Author Contributions:** Conceptualization, F.Á.-Á.; Data curation, M.J.P.-Q.; Formal analysis, F.L.-R., F.Á.-Á. and S.H.; Funding acquisition, S.H.; Investigation, S.H.; Methodology, F.L.-R.; Resources, M.J.P.-Q.; Software, F.L.-R., M.J.P.-Q. and S.H.; Validation, F.Á.-Á.; Visualization, F.Á.-Á.; Writing—original draft, S.H.; Writing—review & editing, F.Á.-Á. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by RTI2018-102256-B-I00 of MINECO/FEDER. The computations of the new simulations were made possible by several generous grants of computing time from the Barcelona Supercomputing Centre, references FI-2018-1-0037, FI-2018-2-0021, FI-2018-3-0032, FI-2019-1-0025, IM-2019-2-2016, AECT-2020-1-0024, AECT-2020-2-0005.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** S.H. is deeply indebted and grateful to his students of the Master of Aerospace Engineering, Universitat Politècnica de València

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| CFD | Compact Finite Differences |
| DNS | Direct Numerical Simulation |
| FD | Finite Differences |
| FFT | Fast Fourier Transform |
| LES | Large Eddy Simulations |
| ODE | Ordinary differential equation |
| PDE | Partial differential equation |
| RANS | Reynolds Averaged Navier Stokes |

## References

1. Pope, S.B. *Turbulent Flows*; Cambridge University Press: Cambridge, UK, 2000.
2. Margot, X.; Hoyas, S.; Fajardo, P.; Patouna, S. A moving mesh generation strategy for solving an injector internal flow problem. *Math. Comput. Model.* **2010**, *52*, 1143–1150. [CrossRef]
3. Margot, X.; Hoyas, S.; Fajardo, P.; Patouna, S. CFD study of needle motion influence on the exit flowconditions of single-hole injectors. *At. Sprays* **2011**, *21*, 31–40. [CrossRef]
4. Escarti-Guillem, M.; Hoyas, S.; García-Raffi, L. Rocket plume URANS simulation using OpenFOAM. *Results Eng.* **2019**, *4*, 100056. [CrossRef]
5. Lluesma-Rodríguez, F.; González, T.; Hoyas, S. CFD simulation of a hyperloop capsule inside a closed environment. *Results Eng.* **2021**, *9*, 100196. [CrossRef]
6. Kolmogorov, A.N. Local structure of turbulence in an incompressible fluid at very high Reynolds numbers. *Dokl. Akad. Nauk.* **1941**, *30*, 9–13. [CrossRef]
7. Kolmogorov, A.N. Dissipation of energy in isotropic turbulence. *Dokl. Akad. Nauk.* **1941**, *32*, 19–21.
8. Samie, M.; Marusic, I.; Hutchins, N.; Fu, M.; Fan, Y.; Hultmark, M.; Smits, A. Fully resolved measurements of turbulent boundary layer flows up to $Re_\tau = 20000$. *J. Fluid Mech.* **2018**, *851*, 391–415. [CrossRef]
9. Kim, J.; Moin, P.; Moser, R. Turbulence statistics in fully developed channels flows at low Reynolds numbers. *J. Fluid Mech.* **1987**, *177*, 133–166. [CrossRef]
10. Spalart, P. Direct simulation of a turbulent boundary layer up to $R_\theta$ = 1410. *J. Fluid Mech.* **1988**, *187*, 61–98. [CrossRef]
11. Moser, R.; Kim, J.; Mansour, N. Direct numerical simulation of turbulent channel flow up to $Re_\tau = 590$. *Phys. Fluids* **1999**, *11*, 943–945. [CrossRef]
12. Hoyas, S.; Jiménez, J. Scaling of the velocity fluctuations in turbulent channels up to $Re_\tau = 2003$. *Phys. Fluids* **2006**, *18*, 011702. [CrossRef]
13. Hoyas, S.; Jiménez, J. Reynolds number effects on the Reynolds-stress budgets in turbulent channels. *Phys. Fluids* **2008**, *20*, 101511. [CrossRef]
14. Simens, M.P.; Jimenez, J.; Hoyas, S.; Mizuno, Y. A high-resolution code for turbulent boundary layers. *J. Comput. Phys.* **2009**, *228*, 4218–4231. [CrossRef]
15. Lee, M.; Moser, R. Direct numerical simulation of turbulent channel flow up to $Re_\tau \approx 5200$. *J. Fluid Mech.* **2015**, *774*, 395–415. [CrossRef]
16. Lozano-Durán, A.; Jiménez, J. Effect of the computational domain on direct simulations of turbulent channels up to $Re_\tau = 4200$. *Phys. Fluids* **2014**, *26*, 011702. [CrossRef]
17. Bernardini, M.; Pirozzoli, S.; Orlandi, P. Velocity statistics in turbulent channel flow up to $Re_\tau = 4000$. *J. Fluid Mech.* **2014**, *758*, 327–343.
18. Avsarkisov, V.; Hoyas, S.; Oberlack, M.; García-Galache, J. Turbulent plane Couette flow at moderately high Reynolds number. *J. Fluid Mech.* **2014**, *751*, R1. [CrossRef]
19. Kraheberger, S.; Hoyas, S.; Oberlack, M. DNS of a turbulent Couette flow at constant wall transpiration up to $Re_\tau = 1000$. *J. Fluid Mech.* **2018**, *835*, 421–443. [CrossRef]
20. Kim, J.; Moin, P. *Transport of Passive Scalars in a Turbulent Channel Flow*; NASA: Ames Research Center: Moffett Field, CA, USA 1987; Volume 1, pp. 1–14.
21. Lyons, S.; Hanratty, T.; McLaughlin, J. Direct numerical simulation of passive heat transfer in a turbulent channel flow. *Int. J. Heat Mass Transf.* **1991**, *34*, 1149–1161. [CrossRef]
22. Kasagi, N.; Tomita, Y.; Kuroda, A. Direct Numerical Simulation of Passive Scalar Field in a Turbulent Channel Flow. *J. Heat Transf.* **1992**, *114*, 598–606. [CrossRef]

23. Piller, M. Direct numerical simulation of turbulent forced convection in a pipe. *Int. J. Numer. Methods Fluids* **2005**, *49*, 583–602. [CrossRef]
24. Yano, T.; Kasagi, N. Direct numerical simulation of turbulent heat transport at high Prandtl numbers. *JSME Int. J. Ser. B Fluids Therm. Eng.* **1999**, *42*, 284–292. [CrossRef]
25. Lluesma-Rodríguez, F.; Hoyas, S.; Peréz-Quiles, M. Influence of the computational domain on DNS of turbulent heat transfer up to $Re_\tau = 2000$ for $Pr = 0.71$. *Int. J. Heat Mass Transf.* **2018**, *122*, 983–992. [CrossRef]
26. Alcántara-Ávila, F.; Hoyas, S.; Pérez-Quiles, M. DNS of thermal channel flow up to $Re_\tau = 2000$ for medium to low Prandtl numbers. *Int. J. Heat Mass Transf.* **2018**, *127*, 349–361. [CrossRef]
27. Alcántara-Ávila, F.; Barberá, G.; Hoyas, S. Evidences of persisting thermal structures in Couette flows. *Int. J. Heat Fluid Flow* **2019**, *76*, 287–295. [CrossRef]
28. Alcántara-Ávila, F.; Hoyas, S. Direct Numerical Simulation of thermal channel flow for medium–high Prandtl numbers up to $Re_\tau = 2000$. *Int. J. Heat Mass Transf.* **2021**, submitted.
29. Alcántara-Ávila, F.; Hoyas, S.; Pérez-Quiles, M. Direct Numerical Simulation of thermal channel fow for $Re_\tau = 5000$ and $Pr = 0.71$. *Accept. J. Fluid Mech.* **2021**.[CrossRef]
30. Gandía Barberá, S.; Alcántara-Ávila, F.; Hoyas, S.; Avsarkisov, V. Stratification effect on extreme-scale rolls in plane Couette flows. *Phys. Rev. E* **2021**, *6*, 034605. [CrossRef]
31. Canuto, C.; Hussaini, M.Y.; Quarteroni, A.M.; Thomas, A., Jr. *Spectral Methods in Fluid Dynamics*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
32. Del Alamo, J.; Jiménez, J.; Zandonade, P.; Moser, R. Scaling of the energy spectra of turbulent channels. *J. Fluid Mech.* **2004**, *500*, 135–144. [CrossRef]
33. Hoyas, S.; Herrero, H.; Mancho, A. Thermocapillar and thermogravitatory waves in a convection problem. *Theor. Comput. Fluid Dyn.* **2004**, *18*, 309–321. [CrossRef]
34. Hoyas, S.; Gil, A.; Fajardo, P.; Pérez-Quiles, M. Codimension-three bifurcations in a Bénard-Marangoni problem. *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.* **2013**, *88*, 015001. [CrossRef] [PubMed]
35. Hoyas, S.; Fajardo, P.; Pérez-Quiles, M. Influence of geometrical parameters on the linear stability of a Bénard-Marangoni problem. *Phys. Rev. E* **2016**, *93*, 043105. [CrossRef] [PubMed]
36. Lele, S.K. Compact finite difference schemes with spectral-like resolution. *J. Comput. Phys.* **1992**, *103*, 16–42. [CrossRef]
37. Orszag, S.A.; Israeli, M. Numerical Simulation of Viscous Incompressible Flows. *Annu. Rev. Fluid Mech.* **1974**, *6*, 281–318. [CrossRef]
38. Rubin, S.; Khosla, P. Polynomial interpolation methods for viscous flow calculations. *J. Comput. Phys.* **1977**, *24*, 217–244. [CrossRef]
39. Spalart, P.R.; Moser, R.D.; Rogers, M.M. Spectral methods for the Navier-Stokes equations with one infinite and two periodic directions. *J. Comput. Phys.* **1991**, *96*, 297–324. [CrossRef]