Article

Two-Dimensional Compact-Finite-Difference Schemes for Solving the bi-Laplacian Operator with Homogeneous Wall-Normal Derivatives

Jesús Amo-Navarro [1], Ricardo Vinuesa [2] , J. Alberto Conejero [1,]* and Sergio Hoyas [1]

[1] Instituto Universitario de Matemática Pura y Aplicada, Universitat Politècnica de València,
    46022 València, Spain; jeamna@etsid.upv.es (J.A.-N.); serhocal@upvnet.upv.es (S.H.)
[2] FLOW, Engineering Mechanics, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden;
    rvinuesa@mech.kth.se
*   Correspondence: aconejero@upv.es

Abstract: In fluid mechanics, the bi-Laplacian operator with Neumann homogeneous boundary conditions emerges when transforming the Navier–Stokes equations to the vorticity–velocity formulation. In the case of problems with a periodic direction, the problem can be transformed into multiple, independent, two-dimensional fourth-order elliptic problems. An efficient method to solve these two-dimensional bi-Laplacian operators with Neumann homogeneus boundary conditions was designed and validated using 2D compact finite difference schemes. The solution is formulated as a linear combination of auxiliary solutions, as many as the number of points on the boundary, a method that was prohibitive some years ago due to the large memory requirements to store all these auxiliary functions. The validation has been made for different field configurations, grid sizes, and stencils of the numerical scheme, showing its potential to tackle high gradient fields as those that can be found in turbulent flows.

Keywords: DNS; CFD; turbulence; bi-Laplacian; fourth-order elliptic

1. Introduction

Turbulence is most likely the open subject in physics with the greatest number of applications in everyday life. Turbulent flows are inherent in practically any flow in engineering, but they are also critical in meteorology and pollutant dispersion [1]. It is known that we still lack an existence and uniqueness theorem for the solution of the Navier–Stokes equations, which govern the behaviour of turbulent flows [2]. Furthermore, apart from a few trivial examples, these equations cannot be solved analytically [1].

It is also well known that turbulent-flow simulations can be considered in terms of three levels of detail: RANS (Reynolds-averaged Navier–Stokes) [3], LES (large-eddy simulation) [4], and DNS (direct numerical simulation) [5,6], each of which corresponds to a different level of accuracy. Nevertheless, until now, DNS has been the only method that has been proven to be reliable for investigating the physical properties of flows that are still poorly understood. On the other hand, DNSs are extremely expensive because every turbulent scale in the flow, both temporal and spatial, must be properly resolved. This requires very fine grids and very small temporal steps. Furthermore, as a result of the evolution of the Kolmogorov scales (the smallest turbulent scale) [7,8], the number of points required to perform the simulation grows extremely quickly, as $Re^{9/4}$, where Re is the Reynolds number. As a result, using DNS to understand turbulent flows requires considering simple geometries, allowing for very precise and fast numerical tools, such as the ones described in this paper. Pipe, channel, and flat-plate boundary layers are examples of canonical wall-bounded flows. A slightly more complex flow case is the square duct [9–11], which is very relevant due to the presence of Prandtl's secondary flow of second kind [12]. The secondary flow in ducts consists of non-vanishing values of the

mean cross-stream velocity components, sharing some similarities with the ones present in Couette flows [13–16]. Due to the presence of the side walls, turbulent ducts are more challenging to simulate [17–19] than channels, pipes, or boundary layers. This method, like many other high-resolution ones, cannot be used in complex geometries. A hole in the domain, for example, would render this method inapplicable. See [20] for details.

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

$$\partial_j U_j \quad = \quad 0, \tag{1}$$

$$\partial_t U_i + U_j \partial_j U_i \quad = \quad -\partial_i P + \frac{1}{Re}\partial_{jj}U_i, \tag{2}$$

where U_j are the velocity components (U, V and W in the streamwise, spanwise and wall-normal directions, respectively), P is the pressure, and Re is the Reynolds number of the problem. Note that in the following x, y, and z are the streamwise, wall-normal and spanwise components, respectively. These equations can be transformed for simple domains [21,22] into the vorticity-Laplacian form, see also [23]. In such geometries, one can take advantage of the derived formulation in the y component of the vorticity ω_y, and the bi-Laplacian of the wall-normal velocity Φ, as explained in Ref. [21]. The main advantages of this method are that the pressure does not need to be explicitly computed, removing staggered grids, and that only these two fields are needed to follow the evolution of the turbulent flow. The equations become:

$$\partial_t \nabla^2 V \quad = \quad -h_v + \frac{1}{Re}\nabla^4 V, \tag{3}$$

$$\partial_t \Omega_y \quad = \quad -h_g + \frac{1}{Re}\nabla^2 \Omega_y, \tag{4}$$

where h_v and h_g collect the nonlinear part of the Equations (1) and (2) [22]. For instance, in turbulent channel flow, there are two periodic directions that allow the use of fast-Fourier methods in these directions. This method was used in the first large DNS of wall turbulence [21], and in many more after that [5,22,24]. Using Fourier methods, instead of solving one large equation one has to solve millions of one-dimensional (1D) equations.

In order to solve these many elliptical problems, compact finite difference (CFD) schemes [25] have been one of the most important tools. Briefly, the main difference between compact and normal finite differences is that the former also considers a stencil in the derivative. Due to their flexibility and high accuracy, CFD schemes have become a very important tool for solving partial-differential-equation problems [26–28].

However, in the case of square (or rectangular) ducts, only the streamwise direction is periodic. Thus, fast-Fourier-transform (FFT) techniques can only be used in this direction, obtaining thousands, instead of millions, of two-dimensional (2D), fourth-order, elliptical problems. Our objective is thus to obtain a method that uses the power and memory of new computers for efficiently solving this problem. Moreover, as it is shown below the boundary conditions (BC) of the problem are both Dirchlet and Neumann homogeneous. We propose an efficient algorithm to impose Neumann BC, taking advantage of the capabilities of CFD to compute any kind of derivatives. Note that our methods are general and can be applied to any other fourth-order elliptic problem, like the bending of beams [29–31].

In the second section, the method is explained in detail: First, we will briefly explain the one-dimensional CFD methods. Then, we explain the discretization of the problem, and the boundary value problem. The validation is detailed in the third section. Finally, conclusions and future work are given in section four.

2. Methods

The geometry of a rectangular duct is shown in Figure 1 (left) together with a sketch of the two-dimensional plane where we solve the bi-Laplacian problem (right panel). The flow is periodic in the streamwise direction and at rest at the walls. The dimensions of the

box are $L_x = D\pi/h$, $L_y = 2h$, and $L_z = 2g$ in the streamwise, wall-normal, and spanwise directions, respectively. D is usually taken large enough to avoid spurious effects due to the periodicity of the box. Equations (3) and (4) are solved together, but in this case, we will focus on Equation (3). Using the auxiliary function $\Phi = \nabla^2 V$, the problem becomes:

$$\partial_t \Phi = -h_v + \frac{1}{Re}\nabla^2\Phi, \tag{5}$$
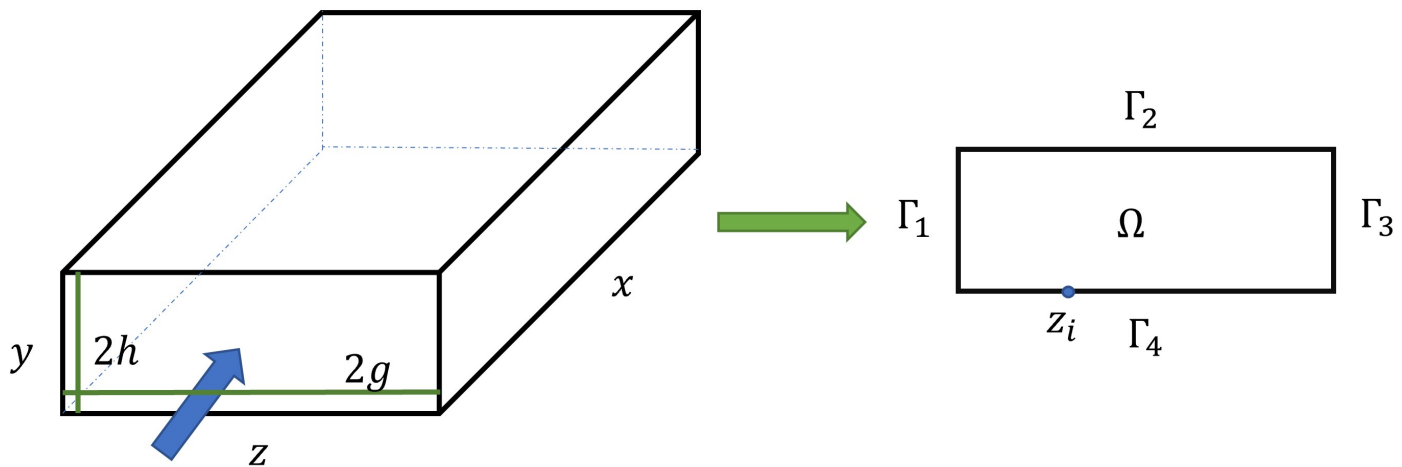
$$\nabla^2 V = \Phi. \tag{6}$$

Figure 1. (**Left**) Sketch of a square duct, and (**Right**) details of the two-dimensional domain.

In Fourier space, these equations transform into n_x problems, where n_x is the size of the transform, which takes the form:

$$\partial_t \hat{\Phi}_k(y,z) = -\hat{h}_V^k + \frac{1}{Re}(-k^2 + \nabla^2)\hat{\Phi}_k(y,z), \tag{7}$$

$$(-k^2 + \nabla^2)\hat{V}_k(y,z) = \hat{\Phi}_k(y,z). \tag{8}$$

In these equations, $\hat{h}_V^k$ is the k-th mode of the Fourier transform of the nonlinear term, which is not usually easy to compute [22]. Note that this allows us to very efficiently distribute these problems in a large supercomputer, as every node has just a few problems to solve, generally one. Applying any time-stepper, with explicit linear solver and implicit non-linear one (see for example Ref. [32]), and removing the hats for simplicity, (7)–(8) transform into:

$$\Phi^{n+1}(y,z) - \alpha\nabla^2\Phi^{n+1}(y,z) = F^n(y,z), \tag{9}$$

$$\nabla^2 V^{n+1}(y,z) = \Phi^{n+1}(y,z). \tag{10}$$

In this equation, $F^n(y,z)$ stands for the RHS of the time stepper [32]. Note that α is a constant, subsuming the Reynolds number, wavenumber k, temporal step Δ, and any constant coming from the temporal-discretization method. For the case of the duct, the boundary conditions for V are Dirichlet and Neumann homogeneous. These boundary conditions come directly from the continuity Equation (1) and the non-slip condition. While the imposition of the former is relatively easy, the latter cannot be imposed directly in the

resolution of the problem. Furthermore, note that there are no boundary conditions for Φ. The final problem, removing the n for simplicity, is then:

$$\Phi(y,z) - \alpha \nabla^2 \Phi(y,z) = F(y,z), \tag{11}$$

$$\nabla^2 V(y,z) = \Phi(y,z), \tag{12}$$

$$V = 0 \text{ in } \partial\Omega = \Gamma_1 + \Gamma_2 + \Gamma_3 + \Gamma_4, \tag{13}$$

$$\partial_n V = 0 \text{ in } \partial\Omega = \Gamma_1 + \Gamma_2 + \Gamma_3 + \Gamma_4. \tag{14}$$

2.1. Compact-Finite-Difference Methods

The CFD method was introduced in a groundbreaking article by S.K. Lele in 1992 [25]. Lele's idea was to use finite-difference methods to solve problems exhibiting a wide range of spatial scales, generalizing some Padé schemes that had been used earlier [33,34]. Lele's CFD main advantage is that, as opposed to to spectral methods, the location of the points is free, exhibiting similar accuracy. The most important fact about CFD methods is that they impose a linear relationship between the values of the derivatives at any order of the function and the value of the function in the vicinity of all points, i.e., for point x_i we have:

$$\sum_{j=-d,d} \alpha_j u^{(l)}(x_{i+j}) = \sum_{j=-f,f} a_j u(x_{i+j}). \tag{15}$$

In this equation, d and f are the semi-size of the stencil, and $u^{(l)}(x_{i+j}) = u^{(l)}_{i+j}$ is the derivative of order l evaluated at x_{i+j}. Using Taylor's theorem as in Ref. [22], it is possible to obtain two sparse matrices such that:

$$A\vec{u}^{(l)} = B\vec{u}. \tag{16}$$

However, it is impossible to repeat this procedure for 2D stencils due to a lack of resolution near the corners of the domain. Thus, the option is to employ 1D discretization in the two directions. Assuming that $u_{i,j} = u(y_i, z_j)$ for $i = 1, ..., n_y$, $j = 1, ..., n_z$ the nodes are distributed in the following way:

$$\mathbf{u} = [u_{i,j}] = \begin{bmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,j} & \cdots & u_{1,n_z} \\ u_{2,1} & \vdots & \cdots & u_{2,j} & \cdots & u_{2,n_z} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ u_{i,1} & u_{i,2} & \cdots & u_{i,j} & \cdots & u_{i,n_z} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ u_{n_y,1} & u_{n_y,2} & \cdots & u_{n_y,j} & \cdots & u_{n_y,n_z} \end{bmatrix}$$

In this equation, the bold face is used to indicate that the data is distributed as a two-dimensional array. Using Equation (16), it is possible to define eight matrices such that:

$$\begin{array}{ll} A_y \mathbf{u}^{(\mathbf{y})} = B_y \mathbf{u}, & A_{yy} \mathbf{u}^{(\mathbf{yy})} = B_{yy} \mathbf{u}, \\ \mathbf{u}^{(\mathbf{z})} (A_z)^t = \mathbf{u}(B_z)^t, & \mathbf{u}^{(\mathbf{zz})} (A_{zz})^t = \mathbf{u}(B_{zz})^t, \end{array} \tag{17}$$

where the superscript indicates a partial derivative in that particular direction.

2.2. Discretization

The discretization of Equations (11) and (12) is similar but, as the former is more general, we will explain only this one. Expanding Equation (11) yields:

$$\mathbf{\Phi} - \alpha(\mathbf{\Phi}^{yy} + \mathbf{\Phi}^{zz}) = \mathbf{F}. \tag{18}$$

Left-multiplying this equation by A_{yy} and right-multiplying it by $(A_{zz})^t$, one gets:

$$A_{yy}\boldsymbol{\Phi} A_{zz}^t - \alpha(A_{yy}\boldsymbol{\Phi}^{(yy)} A_{zz}^t + A_{yy}\boldsymbol{\Phi}^{(zz)} A_{zz}^t) = A_{yy}\mathbf{F} A_{zz}^t. \tag{19}$$

Using now the properties of these CFD matrices, shown in (17), we get

$$A_{yy}\boldsymbol{\Phi}^{(yy)} A_{zz}^t = B_{yy}\boldsymbol{\Phi} A_{zz}^t, \tag{20}$$

$$A_{yy}\boldsymbol{\Phi}^{(zz)} A_{zz}^t = A_{yy}\boldsymbol{\Phi} B_{zz}^t. \tag{21}$$

Using these two equations in (19), the derivatives can be removed from the equation:

$$A_{yy}\boldsymbol{\Phi} A_{zz}^t - \alpha(B_{yy}\boldsymbol{\Phi} A_{zz}^t + A_{yy}\boldsymbol{\Phi} B_{zz}^t) = A_{yy}\mathbf{F} A_{zz}^t. \tag{22}$$

The next step is to transform this problem into a linear equation of the form $A\vec{\phi} = \vec{f}$. The first step is to define an operator, $\otimes$, as

$$G_{AB} \otimes \boldsymbol{\Phi} = A\boldsymbol{\Phi} B^t. \tag{23}$$

Using coordinates we have,

$$[G_{i,j}] \otimes [\Phi_{i,j}] = [B_{i,j}][\Phi_{i,j}][C_{ij}].$$

Defining now $[H_{ij}]$ as

$$[H_{i,j}] = [\Phi_{i,j}][C_{ij}] = \left[\sum_{k=1}^{n_z} \Phi_{i,k} C_{k,j}\right],$$

we obtain:

$$[G_{ij}] \otimes [\Phi_{i,j}] = [B_{ij}][H_{ij}] = \left[\sum_{l=1}^{n_y} B_{il} H_{lj}\right] = \left[\sum_{l=1}^{n_y} B_{il}\left(\sum_{k=1}^{n_z} \Phi_{l,k} C_{k,j}\right)\right].$$

Note that this algorithm can (and must) be simplified, as all the involved matrices are sparse. The problem now becomes

$$G_{A_{yy} A_{zz}} \otimes \boldsymbol{\Phi} - \alpha(G_{B_{yy} A_{zz}} + G_{A_{yy} B_{zz}})\boldsymbol{\Phi} = G_{A_{yy} A_{zz}} \otimes \mathbf{F}. \tag{24}$$

To simplify this equation, we can define an array $\mathbf{M} = G_{A_{yy} A_{zz}} \otimes \mathbf{F}$ and a last operator, G, for the RHS. G is defined as

$$G \otimes \boldsymbol{\Phi} = G_{A_{yy} A_{zz}} \otimes \boldsymbol{\Phi} - \alpha(G_{B_{yy} A_{zz}} + G_{A_{yy} B_{zz}})\boldsymbol{\Phi}, \tag{25}$$

so the problem (21) becomes

$$G \otimes \boldsymbol{\Phi} = \mathbf{M}. \tag{26}$$

Equation (26) still includes the points on the boundary. Given a function $\Psi(y, z)$ that summarises the Dirichlet boundary conditions, the value of $\boldsymbol{\Phi}$ at the boundary is defined as:

$$\boldsymbol{\Phi}^{\partial\Omega} = \begin{bmatrix} \Psi_{1,1} & \Psi_{1,2} & \cdots & \Psi_{1,n_z-1} & \Psi_{1,n_z} \\ \Psi_{2,1} & 0 & \cdots & 0 & \Psi_{2,n_z} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \Psi_{i,1} & 0 & \cdots & 0 & \Psi_{n_y-1,n_z} \\ \Psi_{n_y,1} & \Psi_{n_y,2} & \cdots & \Psi_{n_y,n_z-1} & \Psi_{n_y,n_z} \end{bmatrix},$$

thus, it is possible to decompose $\boldsymbol{\Phi} = \boldsymbol{\Phi}^{\Omega} + \boldsymbol{\Phi}^{\partial\Omega}$ as follows:

$$\mathbf{\Phi} = \begin{bmatrix} \Psi_{1,1} & \Psi_{1,2} & \cdots & \Psi_{1,n_z-1} & \Psi_{1,n_z} \\ \Psi_{2,1} & 0 & \cdots & 0 & \Psi_{2,n_z} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \Psi_{i,1} & 0 & \cdots & 0 & \Psi_{n_y-1,n_z} \\ \Psi_{n_y,1} & \Psi_{n_y,2} & \cdots & \Psi_{n_y,n_z-1} & \Psi_{n_y,n_z} \end{bmatrix} + \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & \phi_{2,2} & \cdots & \phi_{2,n_z-1} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \phi_{n_y-1,2} & \cdots & \phi_{n_y-1,n_z-1} & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

Returning to Equation (26), we have:

$$G \otimes \mathbf{\Phi} = \mathbf{M}, \tag{27}$$

$$G \otimes (\mathbf{\Phi}^{\Omega} + \mathbf{\Phi}^{\partial\Omega}) = \mathbf{M}, \tag{28}$$

$$G \otimes \mathbf{\Phi}^{\Omega} = \mathbf{M} - G \otimes \mathbf{\Phi}^{\partial\Omega} = \mathbf{f}, \tag{29}$$

where only the equations at the inner points make sense. This system can easily be transformed into a sparse algebraic problem of dimension $[(n_y - 2)(n_z - 2), (n_y - 2)(nz - 2)]$.

2.3. The Boundary Problem

The resolution of the problem (11)–(14) is straightforward in the case of Dirichlet boundary conditions. Moreover, the matrix G does not depend on the boundary conditions. The method proposed here generalizes the method proposed in Ref. [21]. The value of the normal derivative is built up using several auxiliary functions. In 1D problems, two auxiliary functions are enough. However, in 2D, many more are needed. Due to this, problems like the square duct have never been addressed using this technique. In this case, the solution is constructed as a linear combination of $2(n_y + n_z) + 1$ fields:

$$V = V_p + c_1 V_1 + c_2 V_2 + \ldots + c_{2n_y+2n_z} V_{2(n_y+n_z)}. \tag{30}$$

The first field in the RHS is the particular solution of the system (11)–(14), with equations:

$$\begin{cases} \Phi - \alpha\nabla^2\Phi = f \\ \nabla^2 V_p = \Phi, \\ V_p = 0 \text{ in } \partial\Omega. \end{cases} \tag{31}$$

Here, V_p is Dirichlet homogeneous, but does not satisfy the Neumann condition. For any point X_l on the boundary, the homogeneous solution V_l is defined as:

$$\begin{cases} \Phi - \alpha\nabla^2\Phi = 0, \\ \Phi_l(X_l) = 1, \\ \nabla^2\Phi_l = \Phi, \\ V_l = 0 \text{ in } \partial\Omega. \end{cases} \tag{32}$$

As V is identically 0 on the boundary, eight of these functions, four corners and two directions, are identically zero, thus only $N_{\text{tot}} = 2n_y + 2n_z - 8$ fields need to be solved. Note that as these fields are solution of a homogeneus equation, we only need to compute them once at the beginning of the program and use them for every time step. This needs a constant time step Δt, as the parameter α usually depends on Δt.

One of the main advantages of the CFD methods is that any derivative can be computed with very high accuracy. This allows us to compute the normal derivatives of the V_l functions:

$$\partial_n V = \partial_n V_p + c_1 \partial_n V_1 + \ldots + c_{N_{\text{tot}}} \partial_n V_{N_{\text{tot}}}. \tag{33}$$

Therefore, we obtain N_{tot} equations, once for each point:

$$\partial_n V(X_l) = \partial_n V_p(X_l) + c_1 \partial_n V_1(X_l) + \ldots + c_{N_{tot}} \partial_n V_{N_{\text{tot}}}(X_l). \tag{34}$$

Applying now the boundary condition and solving for c_l, we obtain:

$$
\begin{bmatrix}
\partial_n u_1(x_1) & \dots & \partial_n u_{N_{tot}}(x_1) \\
\vdots & \vdots & \vdots \\
\partial_n u_1(x_l) & \dots & \partial_n u_{N_{tot}}(x_l) \\
\vdots & \vdots & \vdots \\
\partial_n u_1(x_{N_{tot}}) & \dots & \partial_n u_{N_{tot}}(x_{N_{tot}})
\end{bmatrix}
\begin{bmatrix}
c_1 \\
\vdots \\
c_l \\
\vdots \\
c_{N_{tot}}
\end{bmatrix}
=
\begin{bmatrix}
-\partial_n u_p(x_1) \\
\vdots \\
-\partial_n u_p(x_l) \\
\vdots \\
-\partial_n u_p(x_{N_iot})
\end{bmatrix}.
\tag{35}
$$

The matrix of this system is not sparse anymore, but has a relatively small size. The largest simulations of wall turbulence have around 2000 points in the wall-normal direction [35], so this matrix has a maximum estimated dimension of 8000×8000. This can be solved in seconds in a state-of-the-art computer. Notice also that due to the symmetries of the domain, many of the V_l fields can be computed using these symmetries instead of solving the system (32). In the case of a square domain, only $1/8$ of the fields are needed, as every other can be computed from these ones.

3. Results

The use of high-order numerical schemes such as CFD in fluid dynamics is essential when complex flow structures characterised by high gradients in the magnitudes being calculated appears. Accuracy is then, together with efficiency, the main objective pursued with the numerical method described in this work.

As noted in Section 2.3, the methodology applied to impose the Neumann boundary conditions to a 2D domain and solve the bi-Laplacian operator requires the solution of many linear systems of equations. The only variation among them is the boundary conditions, which end up being part of the right-hand side of the system, remaining the matrix of the system unvaried. The extent to which one can take advantage of this characteristic will significantly determine the computational cost necessary to solve the bi-Laplacian operator. Moreover, the accuracy of the CFD method is defined by the employed stencil, which results in a specific truncation order of the Taylor expansion. The number of non-zero elements of the system matrices changes with the stencil dimension: for a stencil dimension J the number of non-zero elements increases as J^2. As the computational cost increases with the non-zero elements existing in a sparse matrix, a balance between truncation order and computational cost must be found.

Among the different methodologies available to solve the non-symmetric linear systems of equations resulting from the spatial discretization and the implementation of the CFD schemes, the preconditioned generalized minimal residual (GMRES) method has been found to be the most suitable for the study case. This method is included in the SPARSEKIT2 package [36], which contains many preconditioners that vary in computational cost and that fit different matrices according to their conditioning. In this study, the preconditioners that best fit the generated matrices are, on the one hand, the incomplete factorization ILU(0) for preconditioning matrices A_y and A_z. On the other hand, the incomplete LU factorization with standard dropping strategy (ILUD) for the $\overline{G}$ matrices when solving the Laplacian operators. The main advantage of the ILU(0) preconditioner is its simplicity and low computational cost, suitable for matrices with good conditioning. For matrices with worse conditioning, the ILUD preconditioner offers a good ratio convergence/cost. An important advantage is that it provides the possibility to adjust the element-dropping tolerance for prioritizing the preconditioning cost or the convergence speed.

To test the precision with which the system (11)–(14) is being solved, a group of 2D solutions satisfying the Neumann and Dirichlet boundary conditions are set out:

$$
V_{\text{test}}(y, z) = \{\cos[m\pi(y-1)] - 1\}\{\cos[n\pi(z-1)] - 1\},
\tag{36}
$$

where m, n are the function wave numbers.

This product of cosines guarantees that for any $m, n \in \mathbb{Z} \backslash \{0\}$, the value of V_{test} and its first wall-normal derivatives are null. This group of solutions is used to compose a suitable field F_{test}:

$$\Phi_{\text{test}}(y, z) = \nabla^2 V_{\text{test}}(y, z) \tag{37}$$
$$F_{\text{test}}(y, z) = \Phi_{\text{test}}(y, z) - \alpha \nabla^2 \Phi_{\text{test}}(y, z), \tag{38}$$

in such a way that when increasing the wave numbers, the gradients in the solution V_{test} and in the field F_{test} increase as well, see Figure 2.
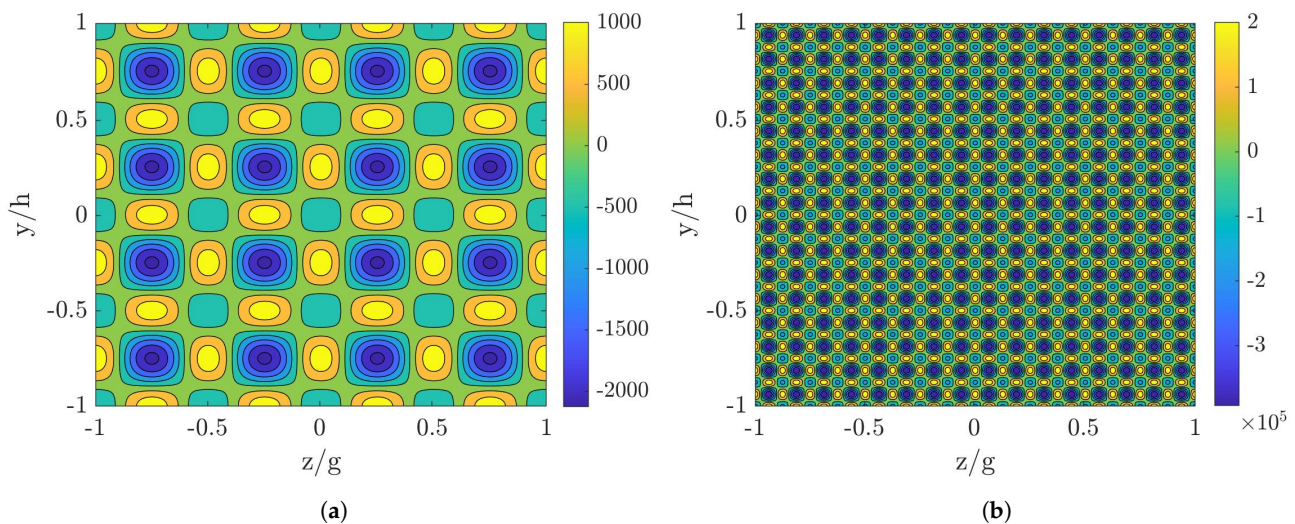
Figure 2. Function F_{test} for (**a**) $m, n = 4$ and (**b**) $m, n = 16$. In both cases, $\alpha = 0.01$.

For sufficiently high wave numbers, V_{test} and F_{test} adopt a chaotic-like distribution similar to that of Gaussian white noise, thus emulating the gradients that could be found in a ducted flow at relatively high Reynolds numbers. System (31) is solved for F_{test} and the numerical solution, V is compared to V_{test}. To analyse this error, two different norms have been used: L2-norm relative error:

$$\|\mathbf{e}\|_2 = \|V - V_{test}\|_2 / \|V_{test}\|_2. \tag{39}$$

The relative error evolution with respect to α, as shown in Figure 3, follows very similar trends among the different wave number, mesh resolution and stencil configurations. In all the configurations proposed, the increment of the relative error with α is significant in the range analysed, with about two orders of magnitude difference between the minimum value of $\alpha = 10^{-7}$ and the maximum one of $\alpha = 10^{-1}$. In this range, there is a perceptible change of tendency; for $\alpha > 10^{-5}$, the trends seem to be linear in a log-log representation, which is indicative of fitting a power function trend, while for $\alpha < 10^{-5}$, the error reduction is less significant. These variations in the errors with α are directly related to the gradients that appear in F_{test} when applying the two Laplacian operators to the imposed solution V_{test}. When reducing the coefficient multiplying the second Laplacian, the effect of this is damped smoothing the gradients in F_{test}.

When comparing grid sizes and truncation orders through the stencil size J, in the range studied, increasing the grid size produces a decrease in the relative error of a similar order of magnitude to that of the increase in the truncation order. However, when comparing the error distributions for $N_y, N_z = 383$; $J = 5$ and $N_y, N_z = 251$; $J = 7$, as the field gradients steepen, the lack of mesh resolution cannot be overcome by the truncation order. However, when the grid resolution is not excessively low with respect to the field gradients, the use of higher-order stencils leads to an important increase in accuracy.
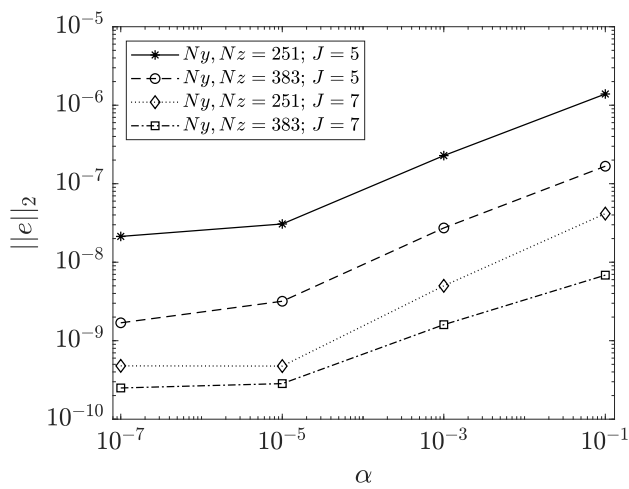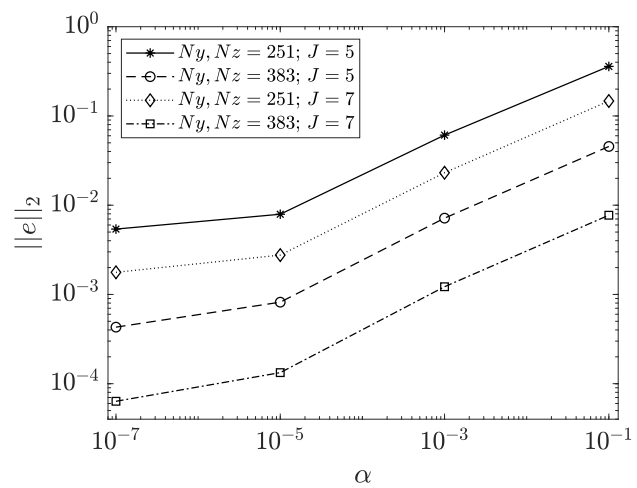
(**a**) $m, n = 4$
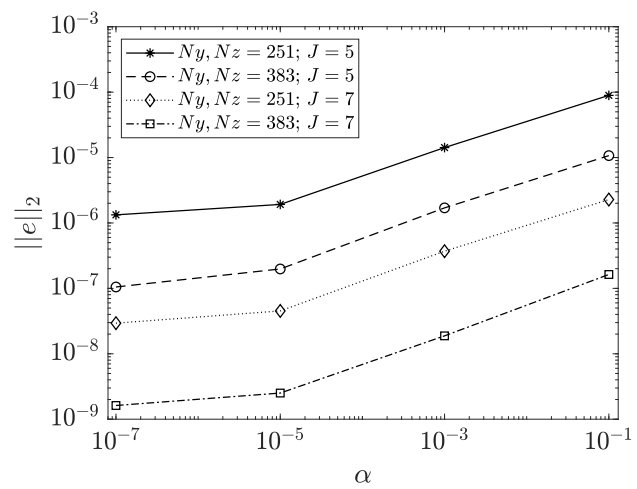
(**b**) $m, n = 8$
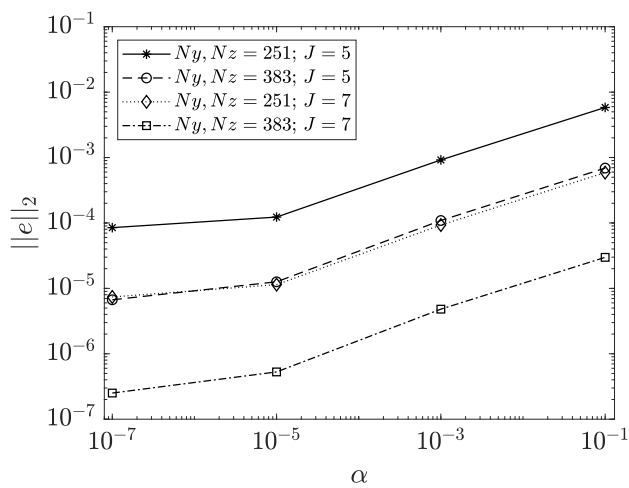
(**c**) $m, n = 16$

(**d**) $m, n = 32$

Figure 3. Evolution of the bi-Laplacian operator solution errors with α.

The spatial distribution of the error is shown in Figure 4. Here, we can see that the error is a heterogeneous field where the higher errors are found mainly in the interior part of the field, decreasing as getting closer to the boundaries. This heterogeneous and non-symmetric error field obtained when solving the bi-Laplacian operator for a symmetric field F_{test} is presumably related to the accuracy with which the system in (35) is solved when obtaining the constants that will weigh each one of the homogeneous solutions. In this regard, a comparison between homogeneous solutions of Φ and V is provided in Figure 5. The damping of the homogeneous solution is much higher for Φ, making imperceptible the transition from the imposed boundary condition to the unaffected near field. In the case of V, the gradients are smaller, extending the contribution of the homogeneous solution all over the field.
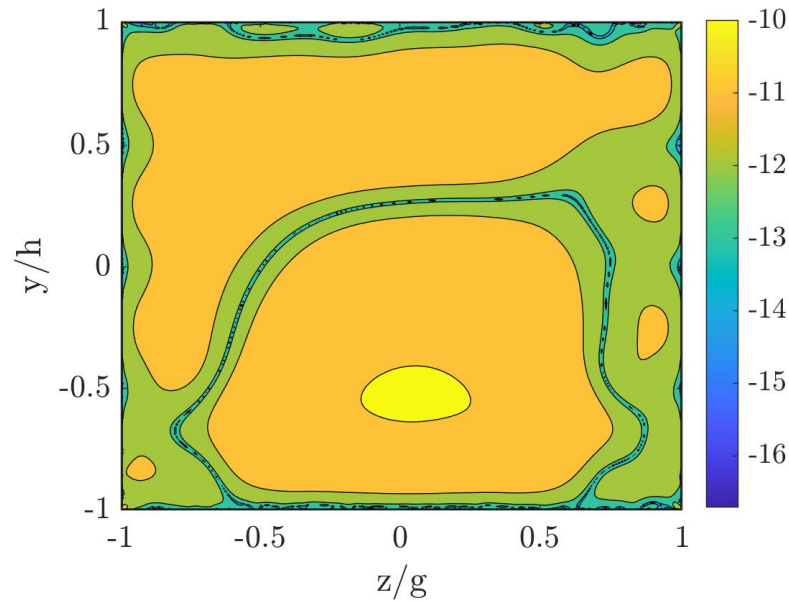
Figure 4. Spatial distribution of the error metric $error = \log_{10}(| V - V_{test} | / \|V_{test}\|_2)$ for a grid$N_y, N_z = 383 \times 383$, $m = 4, n = 6$ and $\alpha = 0.01$.

The high number of homogeneous solutions needed to guarantee that Neumann-homogeneous boundary conditions are satisfied implies saving fields in which a high percentage of elements are close to zero. In order to reduce these memory requirements, that increase with the number of boundary nodes, a filtering of the values that fall below a determined threshold, followed by a sparse storage of the surviving elements of the field could reduce these requirements. However, as shown in field V, the effect of homogeneous solutions spreads all over the field. Therefore a reduction in the accuracy of the method can be expected. In case memory resources constitute an impediment to complete a simulation, deeper studies must be performed to evaluate the accuracy reduction associated to this procedure applied to the specific problem.
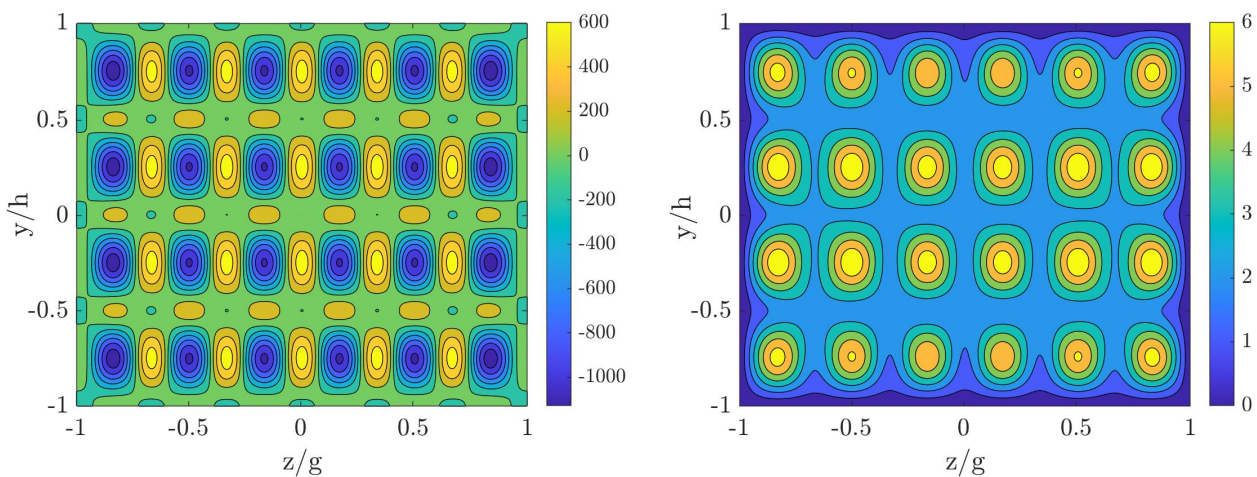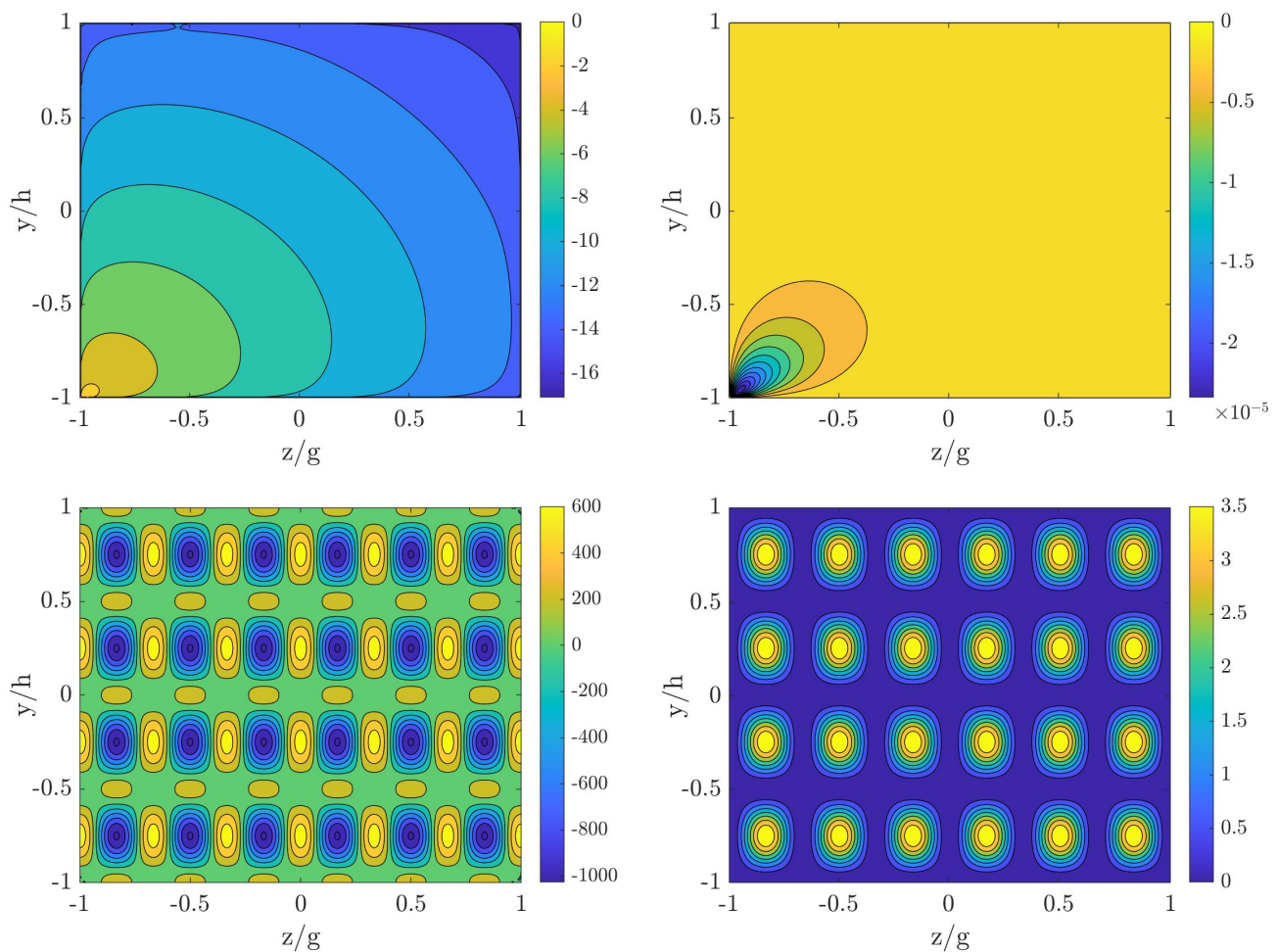

Figure 5. *Cont.*

Figure 5. (Left column) Φ and (right column) V distributions for $\alpha = 0.01$ on a 383×383 grid, for $m = 4, n = 6$. (**Top**) Particular solution, Equation (31); (**Middle**) homogeneus solution, Equation (32) for $X_l = 7$ (the logarithm of Φ is shown to improve readability); and (**Bottom**) solution, Equation (30).

4. Conclusions

In this paper, we have presented a methodology to solve the main equations of a turbulent-flow problem, the turbulent duct. This methodology allows one to solve a fourth-order elliptical problem with Dirichlet and Neumann homogeneous conditions. The main idea behind this method is to take advantage of the high accuracy of compact finite differences to compute derivatives. In the case of temporal integration, all the homogeneous solution can be stored in memory. Then, the bi-Laplacian can be efficiently computed at the cost of two 2D sparse problems and a relatively small full matrix. Thus, a parallel solver for square ducts based on FFT transforms and our method can be efficiently implemented, being a promising alternative for the methods existing right now.

The method has been validated using an ad hoc function presenting an oscillatory behaviour. Using a 7×7 stencil, we are able to solve high oscillatory problems with excellent accuracy. This method can be applied to any fourth-order elliptical problem to solve second- or third-order derivatives at the boundary.

Author Contributions: Conceptualization, R.V., J.A.C. and S.H.; funding acquisition, J.A.C. and S.H.; methodology, J.A.-N., R.V. and J.A.C.; project administration, S.H.; software, J.A.-N.; validation, J.A.-N. and S.H.; visualization, J.A.-N.; writing—original draft, J.A.-N. and S.H.; writing—review and editing, R.V. and J.A.C. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

CFD	Compact finite difference
DNS	Direct numerical simulation
FD	Finite difference
FFT	Fast-Fourier transform
LES	Large-eddy simulation
RANS	Reynolds-averaged Navier–Stokes

References

1. Pope, S.B. *Turbulent Flows*; Cambridge University Press: Cambridge, UK, 2000.
2. Carlson, J.; Wiles, J.; Carlson, J.; Jaffe, A.; Wiles, A.; Institute, C.M.; Society, A.M. *The Millennium Prize Problems*; Amsns AMS Non-Series Title Series; American Mathematical Society: Rhode Island, USA, 2006.
3. Escarti-Guillem, M.; Hoyas, S.; García-Raffi, L. Rocket plume URANS simulation using OpenFOAM. *Results Eng.* **2019**, *4*, 100056.
4. Torres, P.; Le Clainche, S.; Vinuesa, R. On the experimental, numerical and data-driven methods to study urban flows. *Energies* **2021**, *14*, 1310, doi:10.3390/en14051310.
5. Hoyas, S.; Jiménez, J. Scaling of the velocity fluctuations in turbulent channels up to $Re_\tau = 2003$. *Phys. Fluids* **2006**, *18*, 011702.
6. 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**, *176*, 121412.
7. Kolmogorov, A.N. Local structure of turbulence in an incompressible fluid at very high Reynolds numbers. *Dokl. Akad. Nauk.* **1941**, *SSSR (30)*, 9–13.
8. Kolmogorov, A.N. Dissipation of energy in isotropic turbulence. *Dokl. Akad. Nauk.* **1941**, *SSSR (32)*, 19–21.
9. Vinuesa, R.; Schlatter, P.; Nagib, H. Secondary flow in turbulent ducts with increasing aspect ratio. *Phys. Rev. Fluids* **2018**, *3*, 054606, doi:10.1103/PhysRevFluids.3.054606.
10. Pirozzoli, S.; Modesti, D.; Orlandi, P.; Grasso, F. Turbulence and secondary motions in square duct flow. *J. Fluid Mech.* **2018**, *840*, 631–655, doi:10.1017/jfm.2018.66.
11. Gavrilakis, S. Post-transitional periodic flow in a straight square duct. *J. Fluid Mech.* **2019**, *859*, 731–753, doi:10.1017/jfm.2018.811.
12. Prandtl, L. Über die Ausgebildete Turbulenz. In Proceedings of the 2nd International Congress Applied Mechanics, Zurich, Switzerland, 12–17 September 1926; pp. 62–75.
13. Avsarkisov, V.; Oberlack, M.; Hoyas, S. New scaling laws for turbulent Poiseuille flow with wall transpiration. *J. Fluid Mech.* **2014**, *746*, 99–122.
14. Bernardini, M.; Pirozzoli, S.; Orlandi, P. The effect of large-scale turbulent structures on particle dispersion in wall-bounded flows. *Int. J. Multiph. Flow* **2013**, *51*, 55–64.
15. Pirozzoli, S.; Bernardini, M.; Orlandi, P. Turbulence statistics in Couette flow at high Reynolds number. *J. Fluid Mech.* **2014**, *758*, 323–343, doi:10.1017/jfm.2014.529.327.
16. 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, doi:10.1017/jfm.2017.757.
17. Vinuesa, R.; Prus, C.; Schlatter, P.; Nagib, H. Convergence of numerical simulations of turbulent wall-bounded flows and mean cross-flow structure of rectangular ducts. *Meccanica* **2016**, *51*, 3025–3042, doi:10.1007/s11012-016-0558-0.
18. Rodero, C.; Conejero, J.; Garcia-Fernandez, I. Shock wave formation in compliant arteries. *Evol. Equations Control. Theory* **2019**, *8*, 221.
19. Vinuesa, R. High-fidelity simulations in complex geometries: Towards better flow understanding and development of turbulence models. *Results Eng.* **2018**, *11*, 100254.
20. Laizet, S.; Lamballais, E. High-order compact schemes for incompressible flows: A simple and efficient method with quasi-spectral accuracy. *J. Comput. Phys.* **2009**, *228*, 5989–6015, doi:10.1016/j.jcp.2009.05.010.

21. Kim, J.; Moin, P.; Moser, R. Turbulence statistics in fully developed channels flows at low Reynolds numbers. *J. Fluid Mech.* **1987**, *177*, 133–166.
22. Lluesma-Rodríguez, F.; Álcantara Ávila, F.; Pérez-Quiles, M.; Hoyas, S. A code for simulating heat transfer in turbulent channel flow. *Mathematics* **2021**, *9*, 756, doi:10.3390/math9070756.
23. Morteza, S.; Behruz, R. Vortex theory for two dimensional Boussinesq equations. *Appl. Math. Nonlinear Sci.* **2020**, *5*, 67–84, doi:10.2478/amns.2020.2.00014.
24. Lee, M.; Moser, R. Direct numerical simulation of turbulent channel flow up to $Re_\tau \approx 5200$. *J. Fluid Mech.* **2015**, *774*, 395–415.
25. Lele, S.K. Compact finite difference schemes with spectral-like resolution. *J. Comput. Phys.* **1992**, *103*, 16–42.
26. Shukla, R.; Tatineni, M.; Zhong, X. Very high-order compact finite difference schemes on non-uniform grids for incompressible Navier–Stokes equations. *J. Comput. Phys.* **2007**, *224*, 1064–1094, doi:10.1016/j.jcp.2006.11.007.
27. Luo, H.; Baum, J.; Löhner, R. A discontinuous Galerkin method based on a Taylor basis for the compressible flows on arbitrary grids. *J. Comput. Phys.* **2008**, *227*, 8875–8893, doi:10.1016/j.jcp.2008.06.035.
28. Bogey, C.; de Cacqueray, N.; Bailly, C. A shock-capturing methodology based on adaptive spatial filtering for high-order non-linear computations. *J. Comput. Phys.* **2009**, *228*, 1447–1465, doi:10.1016/j.jcp.2008.10.042.
29. Engel, G.; Garikipati, K.; Hughes, T.; Larson, M.; Mazzei, L.; Taylor, R. Continuous/discontinuous finite element approximations of fourth-order elliptic problems in structural and continuum mechanics with applications to thin beams and plates, and strain gradient elasticity. *Comput. Methods Appl. Mech. Eng.* **2002**, *191*, 3669–3750, doi:10.1016/S0045-7825(02)00286-4.
30. Ming, W.; Xu, J. The Morley element for fourth order elliptic equations in any dimensions. *Numer. Math.* **2006**, *103*, 155–169.
31. Ye, Y.; Tang, C.L. Infinitely many solutions for fourth-order elliptic equations. *J. Math. Anal. Appl.* **2012**, *394*, 841–854.
32. Spalart, P.; Moser, R.; Rogers, M. Spectral methods for the Navier-Stokes equations with one infinite and two periodic directions. *J. Comput. Phys.* **1991**, *96*, 297–324.
33. Orszag, S.A.; Israeli, M. Numerical Simulation of Viscous Incompressible Flows. *Annu. Rev. Fluid Mech.* **1974**, *6*, 281–318.
34. Rubin, S.; Khosla, P. Polynomial interpolation methods for viscous flow calculations. *J. Comput. Phys.* **1977**, *24*, 217–244.
35. Hoyas, S.; Oberlack, M.; Kraheberger, S.; Álcantara-Ávila, F.; Laux, J. Wall turbulence at high friction Reynolds numbers. *Phys. Rev. Fluids* **2021**, Submitted.
36. Saad, Y., Iterative Methods for Sparse Linear Systems; SIAM, 2003; pp. 261–281, doi:10.1137/1.9780898718003.ch9. Available online: https://epubs.siam.org/doi/pdf/10.1137/1.9780898718003.ch9 (accessed on 1 June 2021).