



Chapter 13

Turbulence simulation: a case study

Contents

13.1 The incompressible Navier-Stokes equation (NSE)	389
13.1.1 Notation	389
13.1.2 Continuous (PDE) form of the NSE	389
13.1.2.1 Triply periodic case	390
13.1.2.2 Channel flow case	390
13.1.2.3 Duct flow case	391
13.1.2.4 Cavity flow case	391
13.1.3 Conservation properties of the continuous NSE	391
13.1.4 Overall strategy for numerical implementation	391
13.2 Spatial discretization	392
13.2.1 Stretching and staggering of the grid in the wall-bounded direction(s)	392
13.2.2 Second-order finite volume formulations of the spatial derivatives of the NSE	393
13.2.2.1 Channel flow case	393
13.2.2.2 Duct flow case	396
13.2.2.3 Cavity flow case	396
13.2.3 Conservation properties of the spatially-discretized NSE	397
13.2.3.1 Discrete conservation of mass	397
13.2.3.2 Discrete conservation of momentum	398
13.2.3.3 Discrete conservation of energy	399
13.3 Temporal discretization - the fractional step algorithm	402
13.3.1 All viscous terms implicit	403
13.3.2 All y-derivative terms implicit	403
13.3.3 All viscous terms implicit in the triply periodic case	405
13.3.4 All y-derivative terms implicit in the channel-flow case	406
13.3.5 Shared-memory parallelization using OpenMP	408
13.3.6 Distributed-memory parallelization using MPI	408
13.4 Characterizing the statistics of turbulence	408
13.5 The visualization of turbulence	408

13.6 Large eddy simulation	408
13.7 Extensions	408
13.7.1 Passive scalars	408
13.7.2 Active scalars and the Boussinesq approximation of bouyancy	408
13.7.3 Immersed boundary methods	408
13.7.4 Coordinate transformation methods	408
13.7.5 Noise generation	409
13.A Diablo	410
Exercises	410

Chapters 1-10 of this text presented a deliberate sequence of stable, accurate, and efficient numerical methods for a variety of subproblems that must often be addressed in the numerical solution of challenging problems in science and engineering, together with just enough analysis for the computational scientist reading this text to understand the fundamentals of how these algorithms work. Chapter 11 introduced how these methods may be combined in a straightforward manner to simulate some simple PDE systems. We saw that such simulations can quickly challenge the capabilities of the hardware you have at your disposal to perform the computations required. Chapter 12 thus introduced the essential elements of high performance computing necessary to extract the maximum performance from modern computational hardware.

In the present chapter, we synthesize further the methods presented thus far to simulate what is commonly identified as a **computational grand challenge** problem in **computational fluid dynamics (CFD)**, that is, the **direct numerical simulation (DNS)** and **large eddy simulation (LES)** of incompressible flows in simple 3D (or, as a special case, 2D) domains. The case study presented in this chapter should not be considered as an end result of your study of numerical simulation techniques, but rather as an appropriate intermediate step towards the simulation of other large-scale complex systems of interest. The present chapter is included in this text primarily to illustrate how the methods presented thus far may be used efficiently in concert. Some of the significant algorithms leveraged by the code developed in this chapter include those for

- the direct solution of banded linear systems (§2.3),
- the iterative solution of sparse (but not banded) linear systems (§3.2),
- the discretization of spatial derivatives using finite-difference (FD) methods (§8.1),
- the representation of spatial derivatives using spectral methods (§5.2.1), and
- the time marching of an ODE discretization of a PDE system with both linear terms and nonlinear terms using mixed RK/CN methods (§??),

all of which are implemented while paying careful attention to a variety of high-performance computing issues (§12) in order to ensure the resulting numerical code runs efficiently on modern computers. Finally, it is important to note that the code presented here has been developed with a carefully-chosen balance of efficiency, readability, and extensibility in mind, as should any any large-scale simulation code of this sort. That is, when writing a large simulation code, one should always consider both the short-term efficient simulation of the problem(s) at hand and the long-term maintainability of the code developed (i.e., continuing to be able to run it years later, on different computers with different operating systems and installed software libraries), as well as the extensibility of this code, both by the author and by others, to related problems for which it might be well suited. This means documenting the code well, as we have attempted to do both here and in the README text files that accompany the code. These longer-term objectives often take a substantial amount of discipline (both personal and institutional) to complete, especially if the short-term objectives have tight deadlines, which is often the case. The long-term payoff of this discipline can be quite significant.

13.1 The incompressible Navier-Stokes equation (NSE)

13.1.1 Notation

In the development of effective numerical methods for the solution of the incompressible Navier-Stokes equation, we will make use of three distinct notations. Which notation is used in any particular equation is generally self-evident.

In the initial presentation of the continuous Navier-Stokes equation (in §13.1), we use $\{u_1, u_2, u_3\}$ for the velocity components and $\{x_1, x_2, x_3\}$ for the coördinate directions. This facilitates the use of summation notation, e.g., $\partial u_j / \partial x_j \triangleq \partial u_1 / \partial x_1 + \partial u_2 / \partial x_2 + \partial u_3 / \partial x_3$, and saves the letters v and w for different velocity vectors.

In the subsequent presentation and analysis of the spatial discretization of the NSE (in §13.2), the multiple subscripts that would arise following the above notation would get confusing. Thus, we shift in this section to the notation $\{u, v, w\}$ for the velocity individual components, $\{x, y, z\}$ for the coördinate directions, and $\{i, j, k\}$ for gridpoint enumeration.

Finally, in the presentation and discussion of the temporal discretization of the spatially-discretized NSE (in §13.3), we employ the notation $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ for the discretized velocity components. In this notation, the boldface denotes the vector formed by assembling the spatial discretization of the continuous flow variables on all of the gridpoints into a vector, which is in fact how the data is stored in the computer memory. In the computational implementation, it is most convenient to enumerate this vector with a separate index for each coördinate direction (e.g., $u_1(i, j, k)$).

Finally, note that we also make use of the notation \vec{u} (in the spatially-continuous case) and $\vec{\mathbf{u}}$ (in the spatially-discrete case) to denote the collection of all three velocity components; this vector notation extends naturally to $\vec{\psi}, \vec{\phi}, \vec{x}$, etc.

13.1.2 Continuous (PDE) form of the NSE

The equation governing the systems considered in this case study is the incompressible **Navier-Stokes equation (NSE)**, given (in summation notation, for $i \in [1, 2, 3]$ and $j \in [1, 2, 3]$, normalized¹ such that $\rho = 1$) by

$$\frac{\partial u_i}{\partial t} = -\frac{\partial u_j u_i}{\partial x_j} + \mu \frac{\partial^2 u_i}{\partial x_j^2} - \frac{\partial p}{\partial x_i} + \psi_i, \quad (13.1a)$$

$$0 = \frac{\partial u_j}{\partial x_j}, \quad (13.1b)$$

in a 3D (or, as a limiting case, 2D) rectangular domain Ω defined such that $-L_x/2 \leq x \leq L_x/2$, $-L_y/2 \leq y \leq L_y/2$, $-L_z/2 \leq z \leq L_z/2$, with known initial conditions $\vec{u}(t = 0) = \vec{u}_0$, known boundary conditions $\vec{u} = \vec{\phi}$ in 0, 1, 2, or 3 spatial directions, and periodic boundary conditions on the unknowns $\{\vec{u}, p\}$ in the remaining spatial directions. The code described in this chapter will be suitable for the following 4 cases:

- the case with 3 periodic directions, which we will refer to as the **triply periodic** case,
- the case with 2 periodic directions, which we will refer to as the **channel flow** case,
- the case with 1 periodic direction, which we will refer to as the **duct flow** case, and
- the case with 0 periodic directions, which we will refer to as the **cavity flow** case.

¹The assumption that the density $\rho = 1$ in (13.1a) may be relaxed if we replace the pressure p with a symbol understood to denote the **density-normalized pressure**, p/ρ , and if we replace the **dynamic viscosity** μ with the **kinematic viscosity** $\nu = \mu/\rho$. We have not opted to do this in the present chapter, primarily because it is difficult to distinguish the velocity component v and the kinematic viscosity ν in print using the present font.

The first subequation of the NSE², (13.1a), is referred to as the **momentum equation** (with 3 components, one in each coördinate direction), whereas the second subequation of the NSE, (13.1b) is referred to as the **continuity equation**. The momentum equation is an **evolution equation** that is marched in time, whereas the continuity equation is a **constraint equation** that the velocity field must satisfy at each instant. A PDE system of this sort, comprised of both evolution equations and constraint equations, is sometimes referred to as a **differential algebraic equation (DAE)**.

While a significant application in its own right, the Navier-Stokes equation is a valuable problem to consider as a canonical model of PDE systems in general. Though fairly simple to derive and express, Navier-Stokes systems often exhibit chaotic, multiscale dynamics (a.k.a. **turbulence**), the numerical representation of which requires considerable care and attention to a variety of subtle issues to insure the stability, accuracy, and efficiency of the numerical simulation.

13.1.2.1 Triply periodic case

The most fundamental case of interest when looking at turbulence is the decay of unforced **homogenous** (that is, statistically invariant from one *spatial point* to another) **isotropic** (that is, statistically invariant from one *direction* to another) turbulence. In the lab, such homogeneous isotropic turbulence might be approximated by passing a grid through an otherwise quiescent fluid to provide an initial quasi-random agitation, then watching the agitation of this box of turbulence decay. In the cylindrical coördinates of a coffee cup, such an initial quasi-random agitation is commonly provided with a **swizzle stick**.

In order to facilitate the study of the fundamental statistical spectrum of homogeneous isotropic turbulence computationally, one commonly provides some continuous random excitation to a triply periodic flowfield at the largest length scales in the domain (in an unsteady, approximately homogenous, isotropic fashion), then averages in time the statistics of this flow as the energy of it cascades over the higher wavenumbers.

Another case of interest in the triply periodic case is **shear-driven turbulence**. This flow may be studied by replacing \vec{u} with $\vec{U} + \vec{u}$ in the governing equation (13.1), where the background flow profile $U_i(x_1, x_2, x_3) = U_1(x_2)\delta_{i1}$ is specified, then grouping all terms involving \vec{U} into the forcing term $\vec{\Psi}$ and simulating with a minor embellishment of the code. In the simple case that $U_1(x_2) = cx_2$, this flow is homogeneous but not isotropic.

13.1.2.2 Channel flow case

The case of channel flow is the most fundamental realization of wall-bounded turbulence. In the present study, the channel flow case is represented by taking x_2 as the wall-normal direction and by supplying a pressure gradient $\psi_i = -P_x\delta_{i1}$, where $P_x < 0$, forcing the flow in the positive x_1 direction. If the computational box is chosen to be large enough in L_x and L_z (we typically normalize $L_y = 2$ in this case), the (nonphysical) periodic boundary conditions in the x_1 and x_3 directions have minimal effect on the statistics of interest of the near-wall turbulence.

Note that one can also use no-slip BCs at the lower wall and no-shear BCs at the upper wall, together with a “sponge” region (incorporating an artificial “gentle” RHS forcing with minimal upstream influence within the so-called “sponge”) in order to connect a convective outflow condition to a specified inflow profile, thus facilitating the use of a periodic channel flow code to effectively simulate a spatially-developing boundary-layer flow.

²As a reflection of our overall viewpoint of (13.1) defining a single system, we prefer to refer to the NSE in the singular; this viewpoint is reinforced by writing the PDE (13.1) in the form

$$E \frac{\partial \vec{q}}{\partial t} = \mathcal{N}(\vec{q}) \quad \text{where} \quad \vec{q} = \begin{pmatrix} \vec{u} \\ p \end{pmatrix} \quad \text{and} \quad E = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}.$$

13.1.2.3 Duct flow case

The case of duct flow is also easily treated by the present framework. In the present code, the duct flow case is studied by taking x_2 and x_3 as the wall-bounded directions and, as in the channel flow case, supplying a pressure gradient $\psi_i = -P_x \delta_{i1}$ forcing the flow in the positive x_1 direction. Interesting corner effects may be studied in this flow that are relevant to many engineering flows of practical relevance.

13.1.2.4 Cavity flow case

Finally, the present code is also easily extended to the cavity flow case, with all directions bounded by walls and often, to make the flow interesting, nonzero boundary conditions on at least one wall. A canonical problem in this case is given by constantly translating the “lid” (that is, the surface of Ω in the positive x_2 direction) in the x_1 direction, thereby driving a flow within the domain.

13.1.3 Conservation properties of the continuous NSE

Following closely the previous analysis of Burgers’ equation [see (11.37) and the surrounding discussion], it is seen that both the nonlinear and pressure gradient terms of the NSE are energy conserving, whereas the viscous term, which dominates at large wavenumbers, constantly drains energy from the system. To see this, take the RHS forcing $\vec{\psi} = 0$ and boundary conditions $\vec{\phi} = 0$ in any of the 4 cases mentioned above, take the inner product of u_i with (13.1a), integrate over Ω , integrate by parts, and apply (13.1b), which leads to

$$\frac{\partial}{\partial t} \int_{\Omega} \frac{|\vec{u}|^2}{2} d\vec{x} = -\mu \int_{\Omega} |\vec{\nabla} \vec{u}|^2 d\vec{x} \leq 0 \quad \text{where} \quad |\vec{\nabla} \vec{u}|^2 = \sum_{i=1}^3 \sum_{j=1}^3 \left(\frac{\partial u_i}{\partial x_j} \right)^2.$$

Thus, in any of the 4 cases mentioned above, in order for the flow to not decay back to $\vec{u} = 0$, some sort of driving force must be applied (that is, $\vec{\psi} \neq 0$, $\vec{\phi} \neq 0$, or both). Some of the problems of particular interest are outlined above.

Note in particular that the pressure gradient term $\partial p / \partial x_i$ in (13.1a) represents what might be called a **workless force** on the interior of the domain; over any volume, the integral of the force $\partial p / \partial x_i$ times the velocity u_i , via integration by parts and the continuity equation, only picks up a possible contribution from the boundary. To rephrase, the force $\partial p / \partial x_i$ might be said to be **orthogonal to the divergence free manifold** of u_i on the interior of the domain Ω . Yet another way of interpreting this is that the pressure gradient itself may be thought of as a **Lagrange multiplier** (see §21.1.1) which acts to enforce (13.1b) at every instant in the evolution of (13.1a), thereby keeping u_i on this divergence free manifold.

13.1.4 Overall strategy for numerical implementation

To solve the problems described above computationally, the continuous flow field must be discretized on a finite set of points in space, and the PDE governing the flow (that is, the NSE) approximated as a constrained ODE (a.k.a. a **descriptor system**) on this finite set of points. Further, the resulting ODE must be marched in time using discrete time steps. To minimize the expense of the computation, one desires to use as few spatial points as possible and as large time steps as possible while maintaining accuracy (in both space and time) and stability of the simulation. Since the flow is periodic in those directions not bounded by walls, so that there is no inflow or outflow from the domain, it is especially important when the viscosity is made small that numerical errors due to the spatial and temporal discretization of the physical problem do not accumulate in a way which causes the simulation to be unstable. Subject this restriction, a scheme with high spatial accuracy is desired. Finally, with a particular spatial discretization, it is found that certain terms of the governing equation have more restrictive time step limitations than do others in the time-marching algorithm.

The most restrictive terms in any given case should be taken implicitly to allow for stability at “large” time steps (which, however, must be kept small enough to ensure accuracy of the computation), while other less restrictive terms may be taken explicitly. These issues guide the choice of spatial and temporal discretizations of the current problem, which are discussed in detail below.

13.2 Spatial discretization

The spatial discretization used in the present code is a hybrid spectral/FD strategy. For simplicity, all spatial derivatives in all 4 cases are calculated with spectral methods in the spatially periodic direction(s) and with second-order central FD methods in the wall-bounded direction(s).

The numerical grid is chosen to be equispaced and unstaggered in the spatially periodic direction(s), allowing spectral methods (specifically, finite Fourier series expansions) to be used to accurately and efficiently compute all spatial derivatives in these direction(s) [see §5.2.1] at the corresponding gridpoints.

In the wall-bounded direction(s), on the other hand, the grid is chosen to be

- **stretched** (that is, with gridpoints clustered near the boundary of the domain in order to resolve the small-scale flow fluctuations of the flow near the walls), and
- **staggered** (that is, with the various flow variables discretized on sets of points that are offset from one another in order to tightly couple the various flowfield fluctuations governed by the discretized NSE).

Further motivation for and details of this grid stretching and staggering are described in §13.2.1.

Once the (stretched and staggered) grid is defined in the wall-bounded direction(s), a **finite volume** approach³ is proposed to determine approximate expressions for the necessary derivatives of the flow variables in the wall-bounded direction(s), as outlined in §13.2.2.

13.2.1 Stretching and staggering of the grid in the wall-bounded direction(s)

The present code stretches the numerical grid in the wall-bounded direction(s) using a hyperbolic tangent stretching function, as illustrated in Figure 8.1. We will illustrate this stretching (as well as the subsequent staggering) by discussing stretching and staggering of the grid in the y direction, which is the wall-bounded direction in the channel flow case. Note that the duct flow case additionally stretches and staggers the grid in the z direction, and the cavity flow case additionally stretches and staggers the grid in both the x and z directions; both cases will be discussed further at the end of this section.

We initialize the stretched and staggered grid in the channel flow by first defining

$$y_j = \tanh\left(C\left(\frac{2(j-1)}{NY} - 1\right)\right), \quad y_{j+1/2} = \frac{1}{2}(y_j + y_{j+1}). \quad (13.2a)$$

A stretching parameter of $C = 1.75$ is appropriate for many of the flows of interest. The subscript j (for the integer $j \in [0, \dots, NY + 2]$) is used to enumerate what we will call the **base grid**, whereas the subscript $j + 1/2$ (for the integer $j \in [0, \dots, NY + 1]$) is used to enumerate what we will call the **fractional grid** (that is, the set of gridpoints midway between the points in the base grid). Once this grid is appropriately scaled,

- the component of velocity normal to this wall, v , will be discretized on the base grid, whereas
- the components of velocity parallel to this wall, u and w , as well as the pressure, p , will be discretized on the fractional grid.

³That is, a FD approach designed such that certain discrete conservation properties hold via simple telescoping arguments, as discussed further in §13.2.3.

The primary motivation for staggering the wall-normal component of velocity from the other flow variables in this way is to couple the pressure at the nodes with j even to the pressure at the nodes with j odd. This is a natural result of a staggered grid, but is not the case in non-staggered configurations⁴.

Next, we scale this grid to put the desired gridpoints on the wall (at $y = \pm L_y/2$). There are essentially two choices that may be made here: scale the grid such that two planes of the base grid coincide with the wall (e.g., the planes $j = 1$ and $j = NY + 1$), or scale the grid such that two planes from the fractional grid coincide with the wall (e.g., the planes $j = 3/2$ and $j = NY + 1/2$). There are perhaps equally convincing arguments that can be made for either choice. In the present work, we make the latter of these two choices. This choice will be especially convenient when applying the multigrid strategy to the Poisson equation for the pressure update in the fractional step algorithm, to be presented later in this chapter. Selecting $C_1 = 2y_{NY+1/2}/L_y$, we rescale the grid such that

$$y_j \leftarrow y_j/C_1, \quad y_{j+1/2} \leftarrow y_{j+1/2}/C_1 \quad \forall j. \quad (13.2b)$$

Once the grid is rescaled in this fashion, $j = 3/2$ corresponds to the lower wall (at $y = -L_y/2$) and $j = NY + 1/2$ corresponds to the upper wall (at $y = L_y/2$). To simplify the subsequent wall-normal FD calculations, we also make the following definitions:

$$\Delta y_{j+1/2} = y_{j+1} - y_j, \quad \Delta y_j = y_{j+1/2} - y_{j-1/2} = \frac{1}{2} (\Delta y_{j+1/2} + \Delta y_{j-1/2}).$$

For notational convenience in the numerical implementation, which does not allow fractional indices, we also define

$$y_{fj} = y_{j+1/2}, \quad \Delta y_{fj} = \Delta y_{j+1/2}.$$

Also, note that numerical implementation requires standard **ASCII (American Standard Code for Information Interchange)** symbols (that is, non-Greek), so, e.g., Δy and Δy_f are denoted in the code as `DY` and `DYF`. The resulting stretched and staggered grid in a single wall bounded direction, y , as appropriate for the channel flow case, is illustrated in Figure X. Extension to the duct and cavity flow cases is straightforward, as illustrated in Figure Y.

13.2.2 Second-order finite volume formulations of the spatial derivatives of the NSE

13.2.2.1 Channel flow case

To interpolate the flow quantities to the adjacent gridpoints when necessary, the following interpolation formula is used for v

$$\bar{v}_{j+1/2} = \frac{1}{2} (v_{j+1} + v_j) \quad (\text{“fully”-second order}), \quad (13.3)$$

⁴Consider the discretization of the NSE in the channel, duct, or cavity case with a nonstaggered (and unstretched) grid in the wall-bounded direction(s). Label the gridpoints for which the sum of the indices enumerating the FD directions are even as red, and the others as black. Then the discretization of the NSE on each red gridpoint depends only on the pressure at the neighboring black gridpoints, whereas the discretization of the NSE on each black gridpoint depends only on the pressure at the neighboring red gridpoints. That is, the pressure at the red points and the black points are completely decoupled. This can ultimately lead to artificial oscillations in the pressure field in the numerical solution. Gently stretching the grid in the wall-bounded direction(s) fails to alleviate this phenomenon significantly; however, staggering the numerical grid in the FD direction(s) removes this problem completely.

and the following interpolation formulae are used for both u and w (illustrated here for u)

$$\bar{u}_j = \frac{1}{2} (u_{j+1/2} + u_{j-1/2}) \quad (\text{“quasi”-second order}), \quad (13.4)$$

$$\check{u}_j = \frac{1}{2\Delta y_j} (\Delta y_{j+1/2} u_{j+1/2} + \Delta y_{j-1/2} u_{j-1/2}) \quad (\text{“quasi”-second order}), \quad (13.5)$$

$$\check{\check{u}}_j = \frac{1}{2\Delta y_j} (\Delta y_{j-1/2} u_{j+1/2} + \Delta y_{j+1/2} u_{j-1/2}) \quad (\text{“fully”-second order}). \quad (13.6)$$

Interpolation for p is not required in the fractional grid formulation. As, by definition, $y_{j+1/2}$ is midway between y_j and y_{j+1} , the interpolation formula for $\bar{v}_{j+1/2}$ is second-order accurate. As y_j is *not* midway between $y_{j+1/2}$ and $y_{j-1/2}$ due to the grid stretching, only the interpolation formula for $\check{\check{u}}_j$ is truly second-order accurate. The formula for \bar{u}_j and \check{u}_j are only second-order accurate in the sense that, as NY is increased with the stretching function (13.2a) fixed, $\Delta y_{j+1/2}/\Delta y_{j-1/2} \rightarrow 1$, and thus both forms approach a second order form. We will make use of the \bar{u}_j and \check{u}_j interpolation forms in the finite volume formulation that follows.

The motivation for using interpolation forms which are only second order accurate in the sense described above stems from conservation issues, which are described in the following section. Though the “proper” second-order interpolation formula $\check{\check{u}}_j$ could be used everywhere (and, in fact, would be slightly more accurate on a finite grid), the discretization error of such an interpolation formula results in spurious sources and sinks of energy on a marginally-resolved stretched grid, which can lead to numerical instabilities. Proper use of the above interpolation formulae prevents discretization errors from contributing to the total energy of the flow, even on a stretched grid. Note that a sufficiently smooth grid stretching function is used to minimize the inaccuracies caused by these interpolation formulae for reasonable values of NY .

With the spatial discretization of the flow quantities described above, the individual momentum equations are solved at the corresponding velocity points and the continuity equation is enforced on the cells surrounding at the pressure points. The spatial discretization of all derivatives required in the channel case are now made precise. The first component of the momentum equation, (13.1a), to be evaluated at $(i, j + 1/2, k)$ for integer values of i, j , and k , is discretized as follows (suppressing the dependencies of all flow variables on the RHS on i and k for notational clarity)

$$\begin{aligned} \left. \frac{\partial u}{\partial t} \right|_{(i, j+1/2, k)} = & - \left[\frac{\delta_s u_{j+1/2}^2}{\delta x} + \frac{(\bar{v}u)_{j+1} - (\bar{v}u)_j}{\Delta y_{j+1/2}} + \frac{\delta_s (wu)_{j+1/2}}{\delta z} \right] - \frac{\delta_s p_{j+1/2}}{\delta x} + \psi_1 \\ & + \mu \left[\frac{\delta_s^2 u_{j+1/2}}{\delta x^2} + \left(\frac{u_{j+3/2} - u_{j+1/2}}{\Delta y_{j+1}} - \frac{u_{j+1/2} - u_{j-1/2}}{\Delta y_j} \right) / \Delta y_{j+1/2} + \frac{\delta_s^2 u_{j+1/2}}{\delta z^2} \right]. \end{aligned}$$

Note that all derivatives in the x and z directions are computed in Fourier space according to

$$\widehat{\frac{\delta_s q}{\delta x}} = i k_x \hat{q} \quad \widehat{\frac{\delta_s q}{\delta z}} = i k_z \hat{q} \quad \widehat{\frac{\delta_s^2 q}{\delta x}} = -k_x^2 \hat{q} \quad \widehat{\frac{\delta_s^2 q}{\delta z}} = -k_z^2 \hat{q},$$

where the hat indicates the Fourier transform in the x and z directions with corresponding wavenumbers k_x and k_z , q is an arbitrary flow quantity, and the s subscript is used to emphasize that the derivative is evaluated spectrally. Note also that the convective terms involving derivatives in the y direction are computed with “quasi”-second-order accurate FD formulae, whereas the viscous terms involving derivatives in the y direction are evaluated with “fully”-second-order accurate FD formulae. The second component of the momentum equation, evaluated at (i, j, k) for integer values of i, j , and k , is discretized as follows

$$\begin{aligned} \left. \frac{\partial v}{\partial t} \right|_{(i, j, k)} = & - \left[\frac{\delta_s (\check{v}v)_j}{\delta x} + \frac{\bar{v}_{j+1/2}^2 - \bar{v}_{j-1/2}^2}{\Delta y_j} + \frac{\delta_s (\check{w}v)_j}{\delta z} \right] - \frac{p_{j+1/2} - p_{j-1/2}}{\Delta y_j} + \psi_2 \\ & + \mu \left[\frac{\delta_s^2 v_j}{\delta x^2} + \left(\frac{v_{j+1} - v_j}{\Delta y_{j+1/2}} - \frac{v_j - v_{j-1}}{\Delta y_{j-1/2}} \right) / \Delta y_j + \frac{\delta_s^2 v_j}{\delta z^2} \right]. \end{aligned}$$

The third component of the momentum equation, evaluated at $(i, j + 1/2, k)$ for integer values of i, j , and k , is discretized as follows

$$\begin{aligned} \left. \frac{\partial w}{\partial t} \right|_{(i, j+1/2, k)} = & - \left[\frac{\delta_s (uw)_{j+1/2}}{\delta x} + \frac{(\bar{v}\bar{w})_{j+1} - (\bar{v}\bar{w})_j}{\Delta y_{j+1/2}} + \frac{\delta_s w_{j+1/2}^2}{\delta z} \right] - \frac{\delta_s p_{j+1/2}}{\delta z} + \psi_3 \\ & + \mu \left[\frac{\delta_s^2 w_{j+1/2}}{\delta x^2} + \left(\frac{w_{j+3/2} - w_{j+1/2}}{\Delta y_{j+1}} - \frac{w_{j+1/2} - w_{j-1/2}}{\Delta y_j} \right) / \Delta y_{j+1/2} + \frac{\delta_s^2 w_{j+1/2}}{\delta z^2} \right]. \end{aligned}$$

The divergence of the velocity field and the Laplacian of the pressure field, evaluated at $(i, j + 1/2, k)$ for integer values of i, j , and k , is

$$\begin{aligned} & \frac{\delta_s u_{j+1/2}}{\delta x} + \frac{v_{j+1} - v_j}{\Delta y_{j+1/2}} + \frac{\delta_s w_{j+1/2}}{\delta z}, \\ & \frac{\delta_s^2 p_{j+1/2}}{\delta x^2} + \left(\frac{p_{j+3/2} - p_{j+1/2}}{\Delta y_{j+1}} - \frac{p_{j+1/2} - p_{j-1/2}}{\Delta y_j} \right) / \Delta y_{j+1/2} + \frac{\delta_s^2 p_{j+1/2}}{\delta z^2}. \end{aligned}$$

Note that the above two operators are required by the Poisson equation to update the pressure in the fractional step algorithm presented in §13.3.

13.2.2.2 Duct flow case

The duct flow case follows as a straightforward extension of the channel and cavity cases presented above and below, and thus, for brevity, is left as an exercise.

13.2.2.3 Cavity flow case

The spatial discretization of all derivatives required in the cavity case are now made precise. In the following equations, we will need to interpolate flow variables in at most one spatial direction (x , y , or z); the various interpolation formulae described in the first paragraph §13.2.2.1 are thus used again here, replacing Δy with Δx or Δz as appropriate. The first component of the momentum equation, (13.1a), to be evaluated at $\{i, j + 1/2, k + 1/2\}$ for integer values of i , j , and k , is discretized as follows (indicating only the dependencies of flow variables on the RHS on indices other than $\{i, j + 1/2, k + 1/2\}$, for notational clarity)

$$\begin{aligned} \left. \frac{\partial u}{\partial t} \right|_{(i,j+1/2,k+1/2)} = & - \left[\frac{\bar{u}_{i+1/2}^2 - \bar{u}_{i-1/2}^2}{\Delta x_i} + \frac{(\check{v}\bar{u})_{j+1} - (\check{v}\bar{u})_j}{\Delta y_{j+1/2}} + \frac{(\check{w}\bar{u})_{k+1} - (\check{w}\bar{u})_k}{\Delta z_{k+1/2}} \right] - \frac{p_{i+1/2} - p_{i-1/2}}{\Delta x_i} + \psi_1 \\ & + \mu \left[\frac{\frac{u_{i+1} - u_i}{\Delta x_{i+1/2}} - \frac{u_i - u_{i-1}}{\Delta x_{i-1/2}}}{\Delta x_i} + \frac{\frac{u_{j+3/2} - u_{j+1/2}}{\Delta y_{j+1}} - \frac{u_{j+1/2} - u_{j-1/2}}{\Delta y_j}}{\Delta y_{j+1/2}} + \frac{\frac{u_{k+3/2} - u_{k+1/2}}{\Delta z_{k+1}} - \frac{u_{k+1/2} - u_{k-1/2}}{\Delta z_k}}{\Delta z_{k+1/2}} \right]. \end{aligned}$$

The second component of the momentum equation, evaluated at $\{i + 1/2, j, k + 1/2\}$, is discretized as follows (indicating only the dependencies of flow variables on the RHS on indices other than $\{i + 1/2, j, k + 1/2\}$)

$$\begin{aligned} \left. \frac{\partial v}{\partial t} \right|_{(i+1/2,j,k+1/2)} = & - \left[\frac{(\check{u}\bar{v})_{i+1} - (\check{u}\bar{v})_i}{\Delta x_{i+1/2}} + \frac{\bar{v}_{j+1/2}^2 - \bar{v}_{j-1/2}^2}{\Delta y_j} + \frac{(\check{w}\bar{v})_{k+1} - (\check{w}\bar{v})_k}{\Delta y_{k+1/2}} \right] - \frac{p_{j+1/2} - p_{j-1/2}}{\Delta y_j} + \psi_2 \\ & + \mu \left[\frac{\frac{v_{i+3/2} - v_{i+1/2}}{\Delta x_{i+1}} - \frac{v_{i+1/2} - v_{i-1/2}}{\Delta x_i}}{\Delta x_{i+1/2}} + \frac{\frac{v_{j+1} - v_j}{\Delta y_{j+1/2}} - \frac{v_j - v_{j-1}}{\Delta x_{j-1/2}}}{\Delta x_j} + \frac{\frac{v_{k+3/2} - v_{k+1/2}}{\Delta z_{k+1}} - \frac{v_{k+1/2} - v_{k-1/2}}{\Delta z_k}}{\Delta z_{k+1/2}} \right]. \end{aligned}$$

The third component of the momentum equation, evaluated at $\{i + 1/2, j + 1/2, k\}$, is discretized as follows (indicating only the dependencies of flow variables on the RHS on indices other than $\{i + 1/2, j + 1/2, k\}$)

$$\begin{aligned} \left. \frac{\partial w}{\partial t} \right|_{(i+1/2,j+1/2,k)} = & - \left[\frac{(\check{u}\bar{w})_{i+1} - (\check{u}\bar{w})_i}{\Delta x_{i+1/2}} + \frac{(\check{v}\bar{w})_{j+1} - (\check{v}\bar{w})_j}{\Delta y_{j+1/2}} + \frac{\bar{w}_{k+1/2}^2 - \bar{w}_{k-1/2}^2}{\Delta x_k} \right] - \frac{p_{k+1/2} - p_{k-1/2}}{\Delta z_k} + \psi_3 \\ & + \mu \left[\frac{\frac{w_{i+3/2} - w_{i+1/2}}{\Delta x_{i+1}} - \frac{w_{i+1/2} - w_{i-1/2}}{\Delta x_i}}{\Delta x_{i+1/2}} + \frac{\frac{w_{j+3/2} - w_{j+1/2}}{\Delta y_{j+1}} - \frac{w_{j+1/2} - w_{j-1/2}}{\Delta y_j}}{\Delta y_{j+1/2}} + \frac{\frac{w_{k+1} - w_k}{\Delta z_{k+1/2}} - \frac{w_k - w_{k-1}}{\Delta z_{k-1/2}}}{\Delta z_k} \right]. \end{aligned}$$

The divergence of the velocity field and the Laplacian of the pressure field, evaluated at $\{i + 1/2, j + 1/2, k + 1/2\}$, are discretized as follows (indicating only the dependencies of flow variables on points other than $\{i + 1/2, j + 1/2, k + 1/2\}$)

$$\begin{aligned} & \frac{u_{i+1} - u_i}{\Delta x_{i+1/2}} + \frac{v_{j+1} - v_j}{\Delta y_{j+1/2}} + \frac{w_{k+1} - w_k}{\Delta z_{k+1/2}}, \\ & \frac{\frac{p_{i+3/2} - p_{i+1/2}}{\Delta x_{i+1}} - \frac{p_{i+1/2} - p_{i-1/2}}{\Delta x_i}}{\Delta x_{i+1/2}} + \frac{\frac{p_{j+3/2} - p_{j+1/2}}{\Delta y_{j+1}} - \frac{p_{j+1/2} - p_{j-1/2}}{\Delta y_j}}{\Delta y_{j+1/2}} + \frac{\frac{p_{k+3/2} - p_{k+1/2}}{\Delta z_{k+1}} - \frac{p_{k+1/2} - p_{k-1/2}}{\Delta z_j}}{\Delta z_{k+1/2}}. \end{aligned}$$

13.2.3 Conservation properties of the spatially-discretized NSE

An important property that can significantly improve the stability of a nonlinear simulation code is that it conserve as many global properties as possible that the original PDE conserves. We now show, in the channel flow case (even on a stretched grid), that the spatial discretization used in the present code conserves mass to within machine precision, and that errors due to the spatial discretization of the convective terms do not affect the total momentum or energy of the flow. Discrete conservation in the triply periodic, duct, and cavity cases follow as a straightforward extension, and are left as an exercise.

13.2.3.1 Discrete conservation of mass

To show that the total mass is conserved exactly by the spatial discretization in the channel-flow case, the continuity equation is integrated over the volume under consideration, with the integrals evaluated with a rectangular rule in the wall-normal direction and spectral rules in the Fourier directions⁵

$$\begin{aligned} \int_{\Omega} \left(\frac{\delta u}{\delta x} + \frac{\delta v}{\delta y} + \frac{\delta w}{\delta z} \right) dV &= \int_z \int_x \sum_{j=1}^{NY} \Delta y_{j+1/2} \left(\frac{\delta_s u_{j+1/2}}{\delta x} + \frac{v_{j+1} - v_j}{\Delta y_{j+1/2}} + \frac{\delta_s w_{j+1/2}}{\delta z} \right) dx dz \\ &= \int_z \int_x (v_{NY+1} - v_1) dx dz \\ &= 0 \quad \Rightarrow \text{Mass is conserved.} \end{aligned}$$

Note that spectral differentiation in x corresponds to multiplying by ik_x at each wavenumber in Fourier space, whereas spectral integration in x corresponds to picking out the $k_x = 0$ wavenumber; thus, spectral integration of a spectral derivative gives exactly zero. In the wall-normal direction, note that the sum **telescopes**; that is, the positive term for one value of j in the sum exactly cancels the negative term for the next value of j in the sum, so the total sum in the y direction adds up to only a term at each boundary.

⁵For convenience in these expressions, we define v both a half a cell outside the walls and a half a cell inside the walls as the value of v to be prescribed on the boundary, where v is not defined in the staggered discretization; that is,

$$v_{NY+1} = v_{NY} = v_{NY+1/2}, \quad v_1 = v_2 = v_{3/2}.$$

13.2.3.2 Discrete conservation of momentum

To show that total momentum is conserved in each direction in the channel-flow case with no extra forcing (i.e., $\Psi_i = -P_x \delta_{i1}$, $\phi = 0$), each component of the momentum equation in (13.1a) is integrated⁶ over Ω :

$$\begin{aligned}
\frac{\partial}{\partial t} \int_{\Omega} u dV &= \int_{\Omega} \left(-\frac{\delta_s u^2}{\delta x} - \frac{\delta v \bar{u}}{\delta y} - \frac{\delta_s w u}{\delta z} + \mu \left(\frac{\delta_s^2 u}{\delta x^2} + \frac{\delta^2 u}{\delta y^2} + \frac{\delta_s^2 u}{\delta z^2} \right) - \frac{\delta_s p}{\delta x} - P_x \right) dV \\
&= \int_z \int_x \sum_{j=2}^{NY-1} \Delta y_{j+1/2} \left[-\frac{(\bar{v} \bar{u})_{j+1} - (\bar{v} \bar{u})_j}{\Delta y_{j+1/2}} + \frac{\mu}{\Delta y_{j+1/2}} \left(\frac{u_{j+3/2} - u_{j+1/2}}{\Delta y_{j+1}} - \frac{u_{j+1/2} - u_{j-1/2}}{\Delta y_j} \right) - P_x \right] dx dz \\
&= \int_z \int_x \left[\mu \left(\frac{u_{NY+1/2} - u_{NY-1/2}}{\Delta y_{NY}} - \frac{u_{5/2} - u_{3/2}}{\Delta y_2} \right) - L_y P_x \right] dx dz, \\
\frac{\partial}{\partial t} \int_{\Omega} v dV &= \int_{\Omega} \left(-\frac{\delta_s \bar{u} v}{\delta x} - \frac{\delta \bar{v}^2}{\delta y} - \frac{\delta_s \bar{w} v}{\delta z} + \mu \left(\frac{\delta_s^2 v}{\delta x^2} + \frac{\delta^2 v}{\delta y^2} + \frac{\delta_s^2 v}{\delta z^2} \right) - \frac{\delta p}{\delta y} \right) dV \\
&= \int_z \int_x \sum_{j=2}^{NY} \Delta y_j \left[-\frac{\bar{v}_{j+1/2}^2 - \bar{v}_{j-1/2}^2}{\Delta y_j} + \frac{\mu}{\Delta y_j} \left(\frac{v_{j+1} - v_j}{\Delta y_{j+1/2}} - \frac{v_j - v_{j-1}}{\Delta y_{j-1/2}} \right) - \frac{p_{j+1/2} - p_{j-1/2}}{\Delta y_j} \right] dx dz \\
&= \int_z \int_x \left[-p_{NY+1/2} + p_{3/2} \right] dx dz \\
\frac{\partial}{\partial t} \int_{\Omega} w dV &= \int_{\Omega} \left(-\frac{\delta_s u w}{\delta x} - \frac{\delta v \bar{w}}{\delta y} - \frac{\delta_s w^2}{\delta z} + \mu \left(\frac{\delta_s^2 w}{\delta x^2} + \frac{\delta^2 w}{\delta y^2} + \frac{\delta_s^2 w}{\delta z^2} \right) - \frac{\delta_s p}{\delta z} \right) dV \\
&= \int_z \int_x \sum_{j=2}^{NY-1} \Delta y_{j+1/2} \left[-\frac{(\bar{v} \bar{w})_{j+1} - (\bar{v} \bar{w})_j}{\Delta y_{j+1/2}} + \frac{\mu}{\Delta y_{j+1/2}} \left(\frac{w_{j+3/2} - w_{j+1/2}}{\Delta y_{j+1}} - \frac{w_{j+1/2} - w_{j-1/2}}{\Delta y_j} \right) \right] dx dz \\
&= \int_z \int_x \left[\mu \left(\frac{w_{NY+1/2} - w_{NY-1/2}}{\Delta y_{NY}} - \frac{w_{5/2} - w_{3/2}}{\Delta y_2} \right) \right] dx dz
\end{aligned}$$

In the limit that $\mu \rightarrow 0$ with $P_x = 0$, momentum is conserved in the x and z directions. Numerical differencing errors on the interior also do not contribute to a loss of momentum conservation in the y direction; note that, in fact, the numerical code may be implemented in such a manner that $\int_z \int_x v dx dz = 0$ exactly for all y and t . For cases in which $\mu \neq 0$, it is seen that choosing

$$P_x = \frac{1}{V} \int_x \int_z \mu \left(\frac{u_{NY+1/2} - u_{NY-1/2}}{\Delta y_{NY}} - \frac{u_{5/2} - u_{3/2}}{\Delta y_2} \right) dx dz < 0, \quad (13.7)$$

where $V = L_x L_y L_z$ is the volume of the domain under consideration, maintains the x_1 component of momentum (i.e., the **bulk velocity** $u_B = \frac{1}{V} \int_{\Omega} u dV > 0$) constant by balancing the skin friction integrated over the walls with the force applied by the mean pressure gradient⁷.

⁶Note that $du/dt = dw/dt = 0$ on the walls, so the RHS of the u and w components of the momentum equation must be zero at the wall points; thus, these points are skipped in the corresponding sums listed here.

⁷Note that small round-off errors can slowly accumulate to drive the quantity $\frac{1}{V} \int_{\Omega} u dV$ away from the target value of u_B in a long numerical simulation. This can be corrected easily by increasing or decreasing P_x proportionally. An expression of the form

$$P_x = \frac{1}{V} \int_x \int_z \mu \left(\frac{u_{NY+1/2} - u_{NY-1/2}}{\Delta y_{NY}} - \frac{u_{5/2} - u_{3/2}}{\Delta y_2} \right) dx dz + k \left(\frac{1}{V} \int_{\Omega} u dV - u_{B\text{target}} \right),$$

for a small positive gain k (determined by trial and error) is thus usually employed.

13.2.3.3 Discrete conservation of energy

The viscous terms of the NSE result in energy dissipation at the small scales, which is replenished by the action of the pressure gradient P_x on the bulk flow. To show that energy is conserved in cases with forcing $\phi = \psi = 0$ and viscosity $\mu = 0$, the momentum equation in (13.1a) is multiplied by \vec{u} and integrated over the volume under consideration (underbraced sums telescope and therefore cancel):

$$\begin{aligned}
\frac{\partial}{\partial t} \int_{\Omega} \frac{u^2 + v^2 + w^2}{2} dV &= \int_{\Omega} \left[u \left(-\frac{\delta_s u^2}{\delta x} - \frac{\delta v \bar{u}}{\delta y} - \frac{\delta_s w u}{\delta z} + \mu \left(\frac{\delta_s^2 u}{\delta x^2} + \frac{\delta^2 u}{\delta y^2} + \frac{\delta_s^2 u}{\delta z^2} \right) - \frac{\delta_s p}{\delta x} - P_x \right) \right. \\
&\quad + v \left(-\frac{\delta_s \check{u} v}{\delta x} - \frac{\delta \bar{v}^2}{\delta y} - \frac{\delta_s \check{w} v}{\delta z} + \mu \left(\frac{\delta_s^2 v}{\delta x^2} + \frac{\delta^2 v}{\delta y^2} + \frac{\delta_s^2 v}{\delta z^2} \right) - \frac{\delta p}{\delta y} \right) \\
&\quad \left. + w \left(-\frac{\delta_s u w}{\delta x} - \frac{\delta v \bar{w}}{\delta y} - \frac{\delta_s w^2}{\delta z} + \mu \left(\frac{\delta_s^2 w}{\delta x^2} + \frac{\delta^2 w}{\delta y^2} + \frac{\delta_s^2 w}{\delta z^2} \right) - \frac{\delta_s p}{\delta z} \right) \right] dV \\
&= \int_z \int_x \left[\sum_{j=2}^{NY-1} \Delta y_{j+1/2} u_{j+1/2} \left(-\frac{\delta_s u_{j+1/2}^2}{\delta x} - \frac{(v \bar{u})_{j+1} - (v \bar{u})_j}{\Delta y_{j+1/2}} - \frac{\delta_s (w u)_{j+1/2}}{\delta z} - \frac{\delta_s p_{j+1/2}}{\delta x} \right) \right. \\
&\quad + \sum_{j=2}^{NY} \Delta y_j v_j \left(-\frac{\delta_s (\check{u} v)_j}{\delta x} - \frac{\bar{v}_{j+1/2}^2 - \bar{v}_{j-1/2}^2}{\Delta y_j} - \frac{\delta_s (\check{w} v)_j}{\delta z} - \frac{p_{j+1/2} - p_{j-1/2}}{\Delta y_j} \right) \\
&\quad \left. + \sum_{j=2}^{NY-1} \Delta y_{j+1/2} w_{j+1/2} \left(-\frac{\delta_s (u w)_{j+1/2}}{\delta x} - \frac{(v \bar{w})_{j+1} - (v \bar{w})_j}{\Delta y_{j+1/2}} - \frac{\delta_s w_{j+1/2}^2}{\delta z} - \frac{\delta_s p_{j+1/2}}{\delta z} \right) \right] dx dz \\
&= \int_z \int_x \left[\sum_{j=2}^{NY} \Delta y_{j+1/2} \left(p_{j+1/2} - \frac{u_{j+1/2}^2 + (v_{j+1}^2 + v_j^2)/2 + w_{j+1/2}^2}{2} \right) \left(\frac{\delta_s u_{j+1/2}}{\delta x} + \frac{\delta_s w_{j+1/2}}{\delta z} \right) dx dz \right. \\
&\quad - \sum_{j=2}^{NY} \frac{1}{2} \left(\underbrace{u_{j+1/2} u_{j+3/2} v_{j+1} - u_{j-1/2} u_{j+1/2} v_j}_{\text{telescope}} + u_{j+1/2} u_{j+1/2} v_{j+1} - u_{j+1/2} u_{j+1/2} v_j \right) \\
&\quad + \sum_{j=2}^{NY} \Delta y_{j+1/2} p_{j+1/2} \left(\frac{v_{j+1} - v_j}{\Delta y_{j+1/2}} \right) - \sum_{j=2}^{NY-1} \left(\frac{v_{j+1}^2 + v_j^2}{4} (v_{j+1} - v_j) \right) \\
&\quad \left. - \sum_{j=2}^{NY} \frac{1}{2} \left(\underbrace{w_{j+1/2} w_{j+3/2} v_{j+1} - w_{j-1/2} w_{j+1/2} v_j}_{\text{telescope}} + w_{j+1/2} w_{j+1/2} v_{j+1} - w_{j+1/2} w_{j+1/2} v_j \right) \right] dx dz \\
&= \int_z \int_x \sum_{j=2}^{NY} \Delta y_{j+1/2} \left(p_{j+1/2} - \frac{u_{j+1/2}^2 + \frac{v_{j+1}^2 + v_j^2}{2} + w_{j+1/2}^2}{2} \right) \left(\frac{\delta_s u_{j+1/2}}{\delta x} + \frac{v_{j+1} - v_j}{\Delta y_{j+1/2}} + \frac{\delta_s w_{j+1/2}}{\delta z} \right) dx dz \\
&= 0 \quad \Rightarrow \text{energy is conserved.}
\end{aligned}$$

Some additional algebra used in the derivation outlined above now follows. Note first that, in the spectral directions, we may apply what we will call **spectral integration by parts** in our analysis. This property follows from the fact that spectral integration involves simply scaling the $k_x = 0$ mode of the integrand and

noting from (5.33) that

$$\begin{aligned} q = \frac{\delta_s p}{\delta x} u &\Rightarrow \hat{q}_0 = \sum_j (ik_{x_j} \hat{p}_j) \hat{u}_{-j}, \quad \text{and} \\ r = \frac{\delta_s u}{\delta x} p &\Rightarrow \hat{r}_0 = \sum_i (ik_{x_i} \hat{u}_i) \hat{p}_{-i} = \sum_j \hat{u}_{-j} (-ik_{x_j} \hat{p}_j) = -\hat{q}_0. \end{aligned}$$

Thus, in the spectral directions x and z , it follows that

$$\int_x u \frac{\delta_s p}{\delta x} dx = - \int_x \frac{\delta_s u}{\delta x} p dx \quad \text{and} \quad \int_z w \frac{\delta_s p}{\delta z} dz = - \int_z \frac{\delta_s w}{\delta z} p dz.$$

Now consider the spectral derivative of a product, such as $\delta_s uv / \delta x$. Recall that nonlinear products in a pseudospectral code are computed by transforming to physical space, performing the product, then transforming back to Fourier space, whereas spectral differentiation is performed by simply multiplying by ik_x . Recall also that nonlinear products scatter energy to higher wavenumbers. *If* the PDE system under consideration is “fully resolved” in the spectral directions, both products and derivatives would be computed exactly, and we could apply what we will call the **spectral chain rule for differentiation**

$$\frac{\delta_s \check{u}v}{\delta x} = \frac{\delta_s \check{u}}{\delta x} v + \check{u} \frac{\delta_s v}{\delta x}.$$

However, to make them affordable, turbulence simulations are inevitably conducted with “marginal resolution”, using as few Fourier modes as possible in each direction while still achieving the desired accuracy on the quantities of interest in the simulation (typically, some of the time-averaged statistics). Thus, we expect significant energy to cascade to wavenumbers outside the range of wavenumbers represented in the numerical simulation. As discussed in §5.7, there are two primary ways of handling the necessary truncation of the Fourier series representation of the flow field under consideration:

- A) allow the cascade of energy to higher wavenumbers (due to the nonlinear products) to alias back to lower wavenumbers, hoping that the spurious effects of this aliasing will be minimal, or
- B) zero out all higher-order variations resulting from nonlinear products, using the 2/3 dealiasing rule.

Method A creates spurious energy sources, as the spectral chain rule for differentiation shown above does *not* hold when the infinite Fourier series are truncated. Method B constantly drains off the energy of all unresolved modes after each nonlinear product, and thus energy is not conserved in this case either. However, the spectral chain rule for differentiation shown above *does* apply when 2/3 dealiasing is applied to all nonlinear products. Together with the rest of the energy conservation proof provided above, this guarantees that no spurious numerical energy *sources* ever appear in the flow due to the discretization and Fourier series truncation. Thus, in order to insure stability of the computation, we use 2/3 dealiasing in the spectral directions in the present numerical code, acknowledging the extra dissipation that this method applies at the unresolved scales.

The spectral chain rule for differentiation and spectral integration by parts and may thus be applied to the integral of $v \delta_s(uv) / \delta x$ in the spectral direction x , resulting in

$$\int_x v \frac{\delta_s uv}{\delta x} dx = \int_x \left(v^2 \frac{\delta_s u}{\delta x} + uv \frac{\delta_s v}{\delta x} \right) dx = \int_x \left(v^2 \frac{\delta_s u}{\delta x} - v \frac{\delta_s uv}{\delta x} \right) dx \Rightarrow \int_x v \frac{\delta_s uv}{\delta x} dx = \frac{1}{2} \int_x v^2 \frac{\delta_s u}{\delta x} dx.$$

Note that, since the spectral integral of a spectral derivative is zero, it follows from the spectral chain rule for differentiation and the above identity that

$$\int_x \frac{\delta_s u^3}{\delta x} dx = \int_x u \frac{\delta_s u^2}{\delta x} dx = \int_x u^2 \frac{\delta_s u}{\delta x} dx = 0.$$

Note also that, applying the rectangular rule approximation of integration in y and the above identity, we may write

$$\begin{aligned}
\int_x \sum_{j=2}^{NY} \Delta y_j v_j \frac{\delta_s (\check{u}v)_j}{\delta x} dx &= \int_x \sum_{j=2}^{NY} \frac{1}{2} \left(\Delta y_{j+1/2} v_j \frac{\delta_s u_{j+1/2} v_j}{\delta x} + \Delta y_{j-1/2} v_j \frac{\delta_s u_{j-1/2} v_j}{\delta x} \right) dx \\
&= \int_x \sum_{j=2}^{NY} \frac{1}{4} \left(\Delta y_{j+1/2} v_j^2 \frac{\delta_s u_{j+1/2}}{\delta x} + \Delta y_{j-1/2} v_j^2 \frac{\delta_s u_{j-1/2}}{\delta x} \right) dx \\
&= \int_x \sum_{j=2}^{NY-1} \Delta y_{j+1/2} \frac{v_j^2 + v_{j+1}^2}{4} \frac{\delta_s u_{j+1/2}}{\delta x} dx.
\end{aligned}$$

The other convective terms in the spectral directions x and z are handled similarly. Finally, the step involving the rectangular rule approximation of integration of the term $v \delta \bar{v}^2 / \delta y$ may be written

$$\begin{aligned}
\sum_{j=2}^{NY} v_j (\bar{v}_{j+1/2}^2 - \bar{v}_{j-1/2}^2) &= \sum_{j=2}^{NY} \frac{v_j}{4} \underbrace{(v_{j+1} v_j - v_{j-1}^2)}_{+ v_{j+1}^2 - v_j v_{j-1}} + v_{j+1} v_j - v_j v_{j-1} = \sum_{j=2}^{NY} \frac{1}{4} (v_j^2 v_{j+1} - v_j^2 v_{j-1}) \\
&= \sum_{j=2}^{NY} \frac{v_j^2}{4} [(v_{j+1} - v_j) + (v_j - v_{j-1})] = \sum_{j=2}^{NY} \frac{v_j^2 + v_{j+1}^2}{4} (v_{j+1} - v_j).
\end{aligned}$$

13.3 Temporal discretization - the fractional step algorithm

The temporal discretization used in the present work, referred to as a **fractional step** algorithm, is an extension of the mixed RKW3/CN method developed in §10.4.2 applied to the spatial discretization described above of the constrained system (13.1). We first write the time discretization of the spatially-discretized momentum equation (13.1a) as

$$\frac{\mathbf{u}_i^{rk} - \mathbf{u}_i^{rk-1}}{\bar{h}^{rk}} = \bar{\beta}^{rk} \mathbf{r}_i(\bar{\mathbf{u}}^{rk-1}) + \bar{\zeta}^{rk} \mathbf{r}_i(\bar{\mathbf{u}}^{rk-2}) + \frac{1}{2} \left(A_i(\bar{\mathbf{u}}^{rk}) + A_i(\bar{\mathbf{u}}^{rk-1}) \right) - \frac{\delta \mathbf{p}^{rk-1}}{\delta x_i} - \frac{\delta \mathbf{q}}{\delta x_i} + \boldsymbol{\Psi}_i^{rk-1}, \quad (13.8)$$

where $\mathbf{q} \triangleq \mathbf{p}^{rk} - \mathbf{p}^{rk-1}$, and thus pressure is, in effect, treated with IE over each RK substep. In this discretization, $\mathbf{r}_i(\bar{\mathbf{u}})$ represents those (possibly nonlinear) RHS terms of the spatial discretization of (13.1) to be treated explicitly using RKW3 and $A_i(\bar{\mathbf{u}})$ represent those (linear) RHS terms to be treated implicitly using CN over each RK substep; note that there are a couple of possible choices we can make in this regard, as detailed in the following two subsections. The forcing term $\boldsymbol{\Psi}$ is handled with simple EE over each RK substep. The constants \bar{h}^{rk} , $\bar{\beta}^{rk}$, and $\bar{\zeta}^{rk}$ are all defined as in 10.62.

The fractional step algorithm approximates the computation of (13.8) by breaking it into two steps. The first step calculates an intermediate update to the components of the flow velocity, \mathbf{v}_i , neglecting the influence of the pressure update term $-\delta \mathbf{q}/\delta x_i$ on the RHS:

$$\frac{\mathbf{v}_i - \mathbf{u}_i^{rk-1}}{\bar{h}^{rk}} = \bar{\beta}^{rk} \mathbf{r}_i(\bar{\mathbf{u}}^{rk-1}) + \bar{\zeta}^{rk} \mathbf{r}_i(\bar{\mathbf{u}}^{rk-2}) + \frac{1}{2} \left(A_i(\bar{\mathbf{v}}) + A_i(\bar{\mathbf{u}}^{rk-1}) \right) - \frac{\delta \mathbf{p}^{rk-1}}{\delta x_i} + \boldsymbol{\Psi}_i^{rk-1}. \quad (13.9a)$$

The second step then adds the influence of the formerly-neglected pressure update term to the components intermediate velocity field \mathbf{v}_i , and also updates \mathbf{p} itself:

$$\frac{\mathbf{u}_i^{rk} - \mathbf{v}_i}{\bar{h}^{rk}} = -\frac{\delta \mathbf{q}}{\delta x_i}, \quad \mathbf{p}^{rk} = \mathbf{p}^{rk-1} + \mathbf{q}. \quad (13.9b)$$

The pressure update \mathbf{q} is calculated in such a way as to insure that the spatial discretization of the velocity field at the new RK substep, $\bar{\mathbf{u}}^{rk}$, is exactly divergence free, thereby enforcing the spatial discretization of the continuity equation (13.1b). Noting the derivation⁸ in §5.1, this is done simply by defining \mathbf{q} as the solution to the equation

$$\frac{\delta^2 \mathbf{q}}{\delta x_1^2} + \frac{\delta^2 \mathbf{q}}{\delta x_2^2} + \frac{\delta^2 \mathbf{q}}{\delta x_3^2} = \frac{1}{\bar{h}^{rk}} \left(\frac{\delta \mathbf{v}_1}{\delta x_1} + \frac{\delta \mathbf{v}_2}{\delta x_2} + \frac{\delta \mathbf{v}_3}{\delta x_3} \right). \quad (13.10)$$

For convenience, we will apply the same boundary conditions on the intermediate field $\bar{\mathbf{v}}$ and the final field $\bar{\mathbf{u}}^{rk}$; thus, it follows from the normal component of (13.9b) evaluated at the wall(s) that the appropriate boundary conditions on \mathbf{q} on the walls are homogeneous Neumann, i.e., $(\delta \mathbf{q}/\delta n)_{\text{wall}} = 0$. In the directions in which the flow velocity $\bar{\mathbf{u}}$ is periodic, the variables \mathbf{p} and \mathbf{q} are also periodic.

Note that the representation of (13.8) as the two-step process (13.9a)-(13.9b) is only approximate, as we have replaced $A_i(\bar{\mathbf{u}}^{rk})$ on the RHS of (13.8) with $A_i(\bar{\mathbf{v}})$ on the RHS of (13.9a) in order to convert the problem (13.8) to a set of two equations that may be solved one step at a time. As we have been careful to include the explicit term $-\delta \mathbf{p}^{rk-1}/\delta x_i$ in (13.9a), the effect of \mathbf{q} only represents a small “pressure update” over the RK substep, and thus this approximation is, in fact, quite acceptable.

⁸Note that we have selected the constant in this projection as $c = \bar{h}^{rk}$, and thus the velocity update formula given in (5.42c) takes the form shown in (13.9b).

To rearrange and summarize, the equations to be solved by the fractional step algorithm are as follows. First, compute the explicit right-hand sides

$$\mathbf{R}_i = \mathbf{u}_i^{rk-1} + \bar{h}^{rk} \left[\bar{\beta}^{rk} \mathbf{r}_i(\bar{\mathbf{u}}^{rk-1}) + \bar{\zeta}^{rk} \mathbf{r}_i(\bar{\mathbf{u}}^{rk-2}) + \frac{1}{2} A_i(\bar{\mathbf{u}}^{rk-1}) - \frac{\delta \mathbf{p}^{rk-1}}{\delta x_i} + \Psi_i^{rk-1} \right]. \quad (13.11a)$$

Then, solve the implicit systems for the intermediate velocity components \mathbf{v}_i ,

$$\left(I - \frac{\bar{h}^{rk}}{2} A_i \right) \mathbf{v}_i = \mathbf{R}_i, \quad (13.11b)$$

while enforcing the desired boundary conditions for $\bar{\mathbf{u}}^{rk}$ on the intermediate velocity components \mathbf{v}_i . Then solve the Poisson equation for the pressure update \mathbf{q} ,

$$\frac{\delta^2 \mathbf{q}}{\delta x_j^2} = \frac{1}{\bar{h}^{rk}} \frac{\delta \mathbf{v}_j}{\delta x_j}, \quad (13.11c)$$

while enforcing homogeneous Neumann boundary conditions on \mathbf{q} . Finally, update the velocity and pressure accordingly,

$$\mathbf{u}_i^{rk} = \mathbf{v}_i - \bar{h}^{rk} \frac{\delta \mathbf{q}}{\delta x_i}, \quad \mathbf{p}^{rk} = \mathbf{p}^{rk-1} + \mathbf{q}. \quad (13.11d)$$

13.3.1 All viscous terms implicit

As mentioned above, there are a couple of possible choices for which terms to take with RKW3 and which terms to take with CN over each RK substep in the temporal discretization described above. The simplest is to take all (nonlinear) convective terms with RKW3 and all (linear) viscous terms with CN over each RK substep, that is⁹,

$$\mathbf{r}_i(\bar{\mathbf{u}}) = -\frac{\delta \mathbf{u}_1 * \mathbf{u}_i}{\delta x_1} - \frac{\delta \mathbf{u}_2 * \mathbf{u}_i}{\delta x_2} - \frac{\delta \mathbf{u}_3 * \mathbf{u}_i}{\delta x_3}, \quad (13.12a)$$

$$A_i(\bar{\mathbf{u}}) = \mu \left(\frac{\delta^2}{\delta x_1^2} + \frac{\delta^2}{\delta x_2^2} + \frac{\delta^2}{\delta x_3^2} \right) \mathbf{u}_i. \quad (13.12b)$$

This strategy is appropriate when the grid is essentially equally clustered in all three directions.

13.3.2 All y-derivative terms implicit

For systems in which the grid is clustered very tightly in one direction only, such as near the wall in the channel flow case, the wall-normal convective term, if treated explicitly, can result in a significant constraint on the time step for numerical stability. For such systems, it is advantageous to take the wall-normal viscous and wall-normal convective term with CN and all other terms with RKW3, that is⁹,

$$\mathbf{r}_i(\bar{\mathbf{u}}) = \mu \left(\frac{\delta^2 \mathbf{u}_i}{\delta x_1^2} + \frac{\delta^2 \mathbf{u}_i}{\delta x_3^2} \right) - \frac{\delta \mathbf{u}_1 * \mathbf{u}_i}{\delta x_1} - \frac{\delta \mathbf{u}_3 * \mathbf{u}_i}{\delta x_3}, \quad (13.13a)$$

$$A_i(\bar{\mathbf{u}}) = \mu \frac{\delta^2 \mathbf{u}_i}{\delta x_2^2} - \frac{\delta \mathbf{u}_2 * \mathbf{u}_i}{\delta x_2}. \quad (13.13b)$$

⁹The notation $\mathbf{u}_1 * \mathbf{u}_2$ is used to denote the pointwise product of the vector u_1 with the vector u_2 at each gridpoint in physical space.

As this definition of $A_i(\vec{\mathbf{u}})$ is not linear, we can not apply CN to this term directly. However, it is significant to note that, when rearranged properly, this formulation may indeed be approximated by a formulation which is linear in the implicit variables without loss of overall accuracy of the method. To see how this may be accomplished, noting that $(\mathbf{v}_2 - \mathbf{u}_2^{rk-1}) \sim O(h)$ and therefore $(\mathbf{v}_2 - \mathbf{u}_2^{rk-1}) * (\mathbf{v}_2 - \mathbf{u}_2^{rk-1}) \sim O(h^2)$, we note the following identity:

$$0 \approx (\mathbf{v}_2 - \mathbf{u}_2^{rk-1}) * (\mathbf{v}_2 - \mathbf{u}_2^{rk-1}) = \mathbf{v}_2 * \mathbf{v}_2 - 2\mathbf{v}_2 * \mathbf{u}_2^{rk-1} + \mathbf{u}_2^{rk-1} * \mathbf{u}_2^{rk-1} \Rightarrow \mathbf{v}_2 * \mathbf{v}_2 \approx 2\mathbf{v}_2 * \mathbf{u}_2^{rk-1} - \mathbf{u}_2^{rk-1} * \mathbf{u}_2^{rk-1}.$$

Thus, applying this approximation, we may rewrite our temporal discretization of our system, reëxpressing $A_2(\vec{\mathbf{v}})$ as implied by the above identity and moving the extra explicit term so generated onto the RHS, such that¹⁰

$$A_2(\vec{\mathbf{v}}) = \mu \frac{\delta^2 \mathbf{v}_2}{\delta x_2^2} - 2 \frac{\delta \mathbf{u}_2^{rk-1} * \mathbf{v}_2}{\delta x_2}, \quad \mathbf{R}_2 \leftarrow \mathbf{R}_2 + \frac{\bar{h}^{rk}}{2} \frac{\delta \mathbf{u}_2^{rk-1} * \mathbf{u}_2^{rk-1}}{\delta x_2}.$$

Notice that this modified form of $A_2(\vec{\mathbf{v}})$ is indeed linear in the unknown \mathbf{v}_2 , though now the operator A_2 is itself a function of \mathbf{u}_2^{rk-1} . After the \mathbf{v}_2 equation is solved¹¹, we may use \mathbf{v}_2 to approximate \mathbf{u}_2^{rk} in the expressions for $A_1(\vec{\mathbf{u}}^{rk})$ and $A_3(\vec{\mathbf{u}}^{rk})$.

$$A_1(\vec{\mathbf{u}}^{rk}) = \mu \frac{\delta^2 \mathbf{v}_1}{\delta x_2^2} - \frac{\delta \mathbf{v}_2 * \mathbf{v}_1}{\delta x_2}.$$

$$A_3(\vec{\mathbf{u}}^{rk}) = \mu \frac{\delta^2 \mathbf{v}_3}{\delta x_2^2} - \frac{\delta \mathbf{v}_2 * \mathbf{v}_3}{\delta x_2}.$$

Notice that these modified operations are also linear in the unknowns \mathbf{v}_1 and \mathbf{v}_3 respectively, though the linear operators A_1 and A_3 are themselves a function of \mathbf{v}_2 .

¹⁰Fortunately, the new term added to R_2 via this manipulation exactly cancels one of the existing terms of $A_2(\vec{\mathbf{u}}^{rk-1})$ in R_2 [see (13.11a) and (13.12a)], thereby simplifying the computation of R_2 .

¹¹Note that, in this approach, the \mathbf{v}_2 momentum equation must be solved before the \mathbf{v}_1 and \mathbf{v}_3 momentum equations.

13.3.3 All viscous terms implicit in the triply periodic case

In the channel-flow case, all spatial derivatives are calculated spectrally. As introduced in §13.3.1, time stepping in this case may be accomplished with a mixed strategy with all viscous terms treated with CN over each RK substep, and all convective terms treated with RKW3. Each RK substep in this case thus proceeds as follows:

1. Initialize $\widehat{\mathbf{R}}_i$ with $\widehat{\mathbf{u}}^{rk}$ and the explicit part of the CN term

$$\widehat{\mathbf{R}}_i = \left\{ 1 - \frac{\nu}{2} \overline{h}^{rk} (\mathbf{k}_x^2 + \mathbf{k}_y^2 + \mathbf{k}_z^2) \right\} * \widehat{\mathbf{u}}_i$$
2. Account for the pressure gradient term using EE

$$\widehat{\mathbf{R}}_1 \leftarrow \widehat{\mathbf{R}}_1 - \overline{h}^{rk} i\mathbf{k}_x * \widehat{\mathbf{p}}, \quad \widehat{\mathbf{R}}_2 \leftarrow \widehat{\mathbf{R}}_2 - \overline{h}^{rk} i\mathbf{k}_y * \widehat{\mathbf{p}}, \quad \widehat{\mathbf{R}}_3 \leftarrow \widehat{\mathbf{R}}_3 - \overline{h}^{rk} i\mathbf{k}_z * \widehat{\mathbf{p}}.$$
3. Add the RK terms from the previous timestep, stored in $\widehat{\mathbf{F}}_i$, to the RHS

$$\text{if } (rk > 1) \text{ then } \widehat{\mathbf{R}}_i \leftarrow \widehat{\mathbf{R}}_i + \overline{\zeta}^{rk} \overline{h}^{rk} \widehat{\mathbf{F}}_i$$
4. Convert the velocity to physical space

$$\widehat{\mathbf{u}}_i \rightarrow \mathbf{u}_i$$
5. Calculate the nonlinear terms and store in $\widehat{\mathbf{F}}_i$

Note that while there are 9 nonlinear terms, the ordering used here requires only 6 FFT calls.

(a) $\mathbf{F}_1 = \mathbf{u}_1 * \mathbf{u}_1$	(j) $\widehat{\mathbf{F}}_2 \leftarrow -i\mathbf{k}_x * \widehat{\mathbf{F}}_2 - i\mathbf{k}_y * \widehat{\mathbf{S}}$
(b) $\mathbf{F}_2 = \mathbf{u}_1 * \mathbf{u}_2$	(k) $\widehat{\mathbf{F}}_3 \leftarrow -i\mathbf{k}_x * \widehat{\mathbf{F}}_3$
(c) $\mathbf{F}_3 = \mathbf{u}_1 * \mathbf{u}_3$	(l) $\mathbf{S} = \mathbf{u}_2 * \mathbf{u}_3$
(d) $\mathbf{S} = \mathbf{u}_2 * \mathbf{u}_2$	(m) $\mathbf{S} \rightarrow \widehat{\mathbf{S}}$
(e) $\mathbf{F}_1 \rightarrow \widehat{\mathbf{F}}_1$	(o) $\widehat{\mathbf{F}}_2 \leftarrow \widehat{\mathbf{F}}_2 - i\mathbf{k}_z * \widehat{\mathbf{S}}$
(f) $\mathbf{F}_2 \rightarrow \widehat{\mathbf{F}}_2$	(p) $\widehat{\mathbf{F}}_3 \leftarrow \widehat{\mathbf{F}}_3 - i\mathbf{k}_y * \widehat{\mathbf{S}}$
(g) $\mathbf{F}_3 \rightarrow \widehat{\mathbf{F}}_3$	(q) $\mathbf{S} = \mathbf{u}_3 * \mathbf{u}_3$
(h) $\mathbf{S} \rightarrow \widehat{\mathbf{S}}$	(r) $\mathbf{S} \rightarrow \widehat{\mathbf{S}}$
(i) $\widehat{\mathbf{F}}_1 \leftarrow -i\mathbf{k}_x * \widehat{\mathbf{F}}_1 - i\mathbf{k}_y * \widehat{\mathbf{F}}_2 - i\mathbf{k}_z * \widehat{\mathbf{F}}_3$	(s) $\widehat{\mathbf{F}}_3 \leftarrow \widehat{\mathbf{F}}_3 - i\mathbf{k}_z * \widehat{\mathbf{S}}$
6. Add the RK terms from the present timestep to the RHS

$$\widehat{\mathbf{R}}_i \leftarrow \widehat{\mathbf{R}}_i + \overline{\beta}^{rk} \overline{h}^{rk} \widehat{\mathbf{F}}_i$$
7. Solve for the intermediate velocity. Since the system is diagonal, this is easy.

$$\widehat{\mathbf{u}}_i = \widehat{\mathbf{R}}_i / [1 + \frac{\nu}{2} \overline{h}^{rk} (\mathbf{k}_x^2 + \mathbf{k}_y^2 + \mathbf{k}_z^2)]$$
8. Calculate the pressure update \mathbf{q} that will make the velocity divergence free

$$\widehat{\mathbf{q}} = -(\mathbf{i}\mathbf{k}_x \widehat{\mathbf{u}}_1 + \mathbf{i}\mathbf{k}_y \widehat{\mathbf{u}}_2 + \mathbf{i}\mathbf{k}_z \widehat{\mathbf{u}}_3) / (\mathbf{k}_x^2 + \mathbf{k}_y^2 + \mathbf{k}_z^2)$$
9. Project the velocity to get a divergence free field

$$\widehat{\mathbf{u}}_1 \leftarrow \widehat{\mathbf{u}}_1 - i\mathbf{k}_x * \widehat{\mathbf{q}}, \quad \widehat{\mathbf{u}}_2 \leftarrow \widehat{\mathbf{u}}_2 - i\mathbf{k}_y * \widehat{\mathbf{q}}, \quad \widehat{\mathbf{u}}_3 \leftarrow \widehat{\mathbf{u}}_3 - i\mathbf{k}_z * \widehat{\mathbf{q}}.$$
10. Finally, update the pressure field using \mathbf{q}

$$\widehat{\mathbf{p}} \leftarrow \widehat{\mathbf{p}} + \widehat{\mathbf{q}} / \overline{h}^{rk}$$

In all, we have 9 FFT calls per RK substep.

13.3.4 All y-derivative terms implicit in the channel-flow case

In the channel-flow case, the wall-normal (y) derivatives are approximated with FD methods, while the wall-parallel (x and z) derivatives are calculated spectrally. Time-stepping in this case may be accomplished with either a slight modification of the all-viscous-terms-implicit approach described above, or, as introduced in §13.3.2, an alternative mixed strategy with the viscous and convective terms involving wall-normal derivatives treated with CN over each RK substep, and the remaining viscous and convective terms treated with RKW3. Following the spatial discretization used in §13.2.2.1, each RK substep in this case proceeds as follows:

1. Initialize $\widehat{\mathbf{R}}_i$ with $\widehat{\mathbf{u}}^{rk}$
 $\widehat{\mathbf{R}}_i = \widehat{\mathbf{u}}_i$
2. Account for the pressure gradient term using EE
 $\widehat{\mathbf{R}}_1 \leftarrow \widehat{\mathbf{R}}_1 - \bar{h}^{rk} \mathbf{i} \mathbf{k}_x * \widehat{\mathbf{p}}, \quad \widehat{\mathbf{R}}_2 \leftarrow \widehat{\mathbf{R}}_2 - \bar{h}^{rk} (\widehat{\mathbf{p}}_x, \quad \widehat{\mathbf{R}}_3 \leftarrow \widehat{\mathbf{R}}_3 - \bar{h}^{rk} \mathbf{i} \mathbf{k}_z * \widehat{\mathbf{p}}.$
3. If ($RK > 1$) then add the term from the previous RK step
 $\widehat{\mathbf{R}}_i = \widehat{\mathbf{R}}_i + \bar{\zeta}_{RK} \bar{\beta}_{RK} \widehat{\mathbf{F}}_i$
4. Add the pressure gradient to the RHS
 $\widehat{\mathbf{R}}_1 = \widehat{\mathbf{R}}_1 - \bar{h}_{RK} \mathbf{i} k_x \widehat{P}$
 $\widehat{\mathbf{R}}_2(k_x, k_z, j) = \widehat{\mathbf{R}}_2(k_x, k_z, j) - \bar{h}_{RK} \frac{\widehat{P}(k_x, k_z, j) - \widehat{P}(k_x, k_z, j-1)}{\Delta Y(j)}$
 $\widehat{\mathbf{R}}_3 = \widehat{\mathbf{R}}_3 - \bar{h}_{RK} \mathbf{i} k_z \widehat{P}$
5. Add P_x , the background pressure gradient that drives the flow
 $\widehat{\mathbf{R}}_1(k_x = 0, k_z = 0, j) = \widehat{\mathbf{R}}_1(k_x = 0, k_z = 0, j) - \bar{h}_{RK} P_x$
6. Create a storage variable \mathbf{F} that will contain all RK terms and start with the viscous terms involving horizontal derivatives.
 $\widehat{\mathbf{F}}_i = -\nu(k_x^2 + k_z^2) \widehat{\mathbf{u}}_i,$
7. Convert the velocity to physical space
 $\widehat{\mathbf{u}}_i \rightarrow \mathbf{u}_i$
8. Add the nonlinear terms involving horizontal derivatives to $\widehat{\mathbf{F}}$
 $\widehat{\mathbf{F}}_1 = \widehat{\mathbf{F}}_1 - \mathbf{i} k_x \widehat{\mathbf{u}}_1 \widehat{\mathbf{u}}_3 - \mathbf{i} k_x \widehat{\mathbf{u}}_1 \widehat{\mathbf{u}}_1$
 $\widehat{\mathbf{F}}_2 = \widehat{\mathbf{F}}_2 - \mathbf{i} k_x \widehat{\mathbf{u}}_1 \widehat{\mathbf{u}}_2 - \mathbf{i} k_z \widehat{\mathbf{u}}_3 \widehat{\mathbf{u}}_2$
 $\widehat{\mathbf{F}}_3 = \widehat{\mathbf{F}}_3 - \mathbf{i} k_z \widehat{\mathbf{u}}_1 \widehat{\mathbf{u}}_3 - \mathbf{i} k_z \widehat{\mathbf{u}}_3 \widehat{\mathbf{u}}_3$
 (Note that we need 5 independent FFTs here)
9. Now, we are done building the Runge-Kutta terms, add to the right hand side. We will need to keep $\widehat{\mathbf{F}}_i$ for the next RK step, so it should not be overwritten below this point.
 $\widehat{\mathbf{R}}_i = \widehat{\mathbf{R}}_i + \bar{\beta}_{RK} \bar{h}_{RK} \widehat{\mathbf{F}}_i$
10. Convert the right hand side arrays to physical space
 $\widehat{\mathbf{R}}_i \rightarrow \mathbf{R}_i,$
11. Compute the vertical viscous terms and add to the RHS as the explicit part of Crank-Nicolson.

$$R_1(i, j, k) = R_1(i, j, k) + \frac{\bar{\nu} \bar{h}_{RK}}{2} \left(\frac{u_1(i, j+1, k) - u_1(i, j, k)}{\Delta Y(j+1)} - \frac{u_1(i, j, k) - u_1(i, j-1, k)}{\Delta Y(j)} \right) / \Delta Y_F(j)$$

$$R_2(i, j, k) = R_2(i, j, k) + \frac{\bar{\nu} \bar{h}_{RK}}{2} \left(\frac{u_2(i, j+1, k) - u_2(i, j, k)}{\Delta Y_F(j)} - \frac{u_2(i, j, k) - u_2(i, j-1, k)}{\Delta Y_F(j-1)} \right) / \Delta Y(j)$$

$$R_3(i, j, k) = R_3(i, j, k) + \frac{\bar{v}h_{RK}}{2} \left(\frac{u_3(i, j+1, k) - u_3(i, j, k)}{\Delta Y(j+1)} - \frac{u_3(i, j, k) - u_3(i, j-1, k)}{\Delta Y(j)} \right) / \Delta Y_F(j)$$

12. Compute the nonlinear terms involving vertical derivatives and add to the RHS as the explicit part of Crank-Nicolson.

$$S_1 = \bar{u}_1 * u_2$$

$$R_1(i, j, k) = R_1(i, j, k) - \frac{\bar{h}_{RK}}{2} (S_1(i, j+1, k) - S_1(i, j, k)) / \Delta Y_F(j)$$

$$S_1 = \bar{u}_3 * u_2$$

$$R_3(i, j, k) = R_3(i, j, k) - \frac{\bar{h}_{RK}}{2} (S_1(i, j+1, k) - S_1(i, j, k)) / \Delta Y_F(j)$$

13. Solve the tridiagonal system for the intermediate wall-normal velocity:

$$v_2(i, j, k) - \frac{\bar{v}h_{RK}}{2} \left(\frac{v_2(i, j+1, k) - v_2(i, j, k)}{\Delta Y_F(j)} - \frac{v_2(i, j, k) - v_2(i, j-1, k)}{\Delta Y_F(j-1)} \right) / \Delta Y(j) \\ + \bar{h}_{RK} (\bar{v}_2(i, j, k)\bar{u}_2(i, j, k) - \bar{v}_2(i, j-1, k)\bar{u}_2(i, j-1, k)) / \Delta Y(j) = R_2(i, j, k)$$

14. Now that we have the new intermediate wall-normal velocity, v_2 , solve for the intermediate v_1 and v_3 using this new velocity.

$$v_1(i, j, k) - \frac{\bar{v}h_{RK}}{2} \left(\frac{v_1(i, j+1, k) - v_1(i, j, k)}{\Delta Y(j+1)} - \frac{v_1(i, j, k) - v_1(i, j-1, k)}{\Delta Y(j)} \right) / \Delta Y_F(j) \\ + \bar{h}_{RK} (\bar{v}_1(i, j+1, k)v_2(i, j+1, k) - \bar{v}_1(i, j, k)v_2(i, j, k)) / \Delta Y_F(j) = R_1(i, j, k)$$

$$v_3(i, j, k) - \frac{\bar{v}h_{RK}}{2} \left(\frac{v_3(i, j+1, k) - v_3(i, j, k)}{\Delta Y(j+1)} - \frac{v_3(i, j, k) - v_3(i, j-1, k)}{\Delta Y(j)} \right) / \Delta Y_F(j) \\ + \bar{h}_{RK} (\bar{v}_3(i, j+1, k)v_2(i, j+1, k) - \bar{v}_3(i, j, k)v_2(i, j, k)) / \Delta Y_F(j) = R_3(i, j, k)$$

15. Convert the intermediate velocity to Fourier space

$$v_i \rightarrow \hat{v}_i$$

16. Solve the tridiagonal system for the pressure correction:

$$-(k_x^2 + k_z^2)\hat{q}(k_x, k_z, j) + \left(\frac{\hat{q}(k_x, k_z, j+1) - \hat{q}(k_x, k_z, j)}{\Delta Y(j+1)} - \frac{\hat{q}(k_x, k_z, j) - \hat{q}(k_x, k_z, j-1)}{\Delta Y(j)} \right) / \Delta Y_F(j) \\ = \hat{i}k_x \hat{v}_1(k_x, k_z, j) + \hat{i}k_z \hat{v}_3(k_x, k_z, j) + (\hat{v}_2(k_x, k_z, j+1) - \hat{v}_2(k_x, k_z, j)) / \Delta Y_F(j)$$

(Note that in order to avoid an extra storage array, we can store q in R_1 which is no longer needed for this RK step. Also notice that a factor of \bar{h}_{RK} has been absorbed into q)

17. Now, use the pressure update to obtain a divergence-free velocity field.

$$\hat{u}_1^{RK+1} = \hat{v}_1 - \hat{i}k_x \hat{q}$$

$$\hat{u}_2^{RK+1}(k_x, k_z, j) = \hat{v}_2(k_x, k_z, j) - (\hat{q}(k_x, k_z, j) - \hat{q}(k_x, k_z, j-1)) / \Delta Y(j)$$

$$\hat{u}_3^{RK+1} \hat{v}_3 - \hat{i}k_z \hat{q}$$

(In order to avoid an extra storage array, only one set of velocity arrays are defined, and this update is done in place.)

18. Finally, update the pressure field using q

$$\hat{P} = \hat{P} + \hat{q}/\bar{h}_{RK}$$

(We need to divide by \bar{h}_{RK} since this constant has been absorbed into \bar{q} in the steps above.)

In all, we have 14 FFT calls per Runge-Kutta substep, and 11 full-sized storage arrays.

13.3.5 Shared-memory parallelization using OpenMP

13.3.6 Distributed-memory parallelization using MPI

13.4 Characterizing the statistics of turbulence

13.5 The visualization of turbulence

The discriminant.

Data Explorer

13.6 Large eddy simulation

13.7 Extensions

13.7.1 Passive scalars

13.7.2 Active scalars and the Boussinesq approximation of bouyancy

Introduce the compressible Navier-Stokes equation. Discuss computational issues. Define Mach number. Introduce Boussinesq.

13.7.3 Immersed boundary methods

13.7.4 Coordinate transformation methods

13.7.5 Noise generation

Neglecting viscosity, the equation governing compressible flow (known as the **inviscid Euler equation**), in conservation form, is

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0, \quad (13.14a)$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_j u_i}{\partial x_j} + \frac{\partial p}{\partial x_i} = 0. \quad (13.14b)$$

Subtracting $\partial^2 \rho / \partial t^2$ from $\partial^2 p / \partial x_i^2$ and applying $\partial / \partial t$ of (13.14a) then (13.14b), we obtain

$$\frac{\partial^2 p}{\partial x_i^2} - \frac{\partial^2 \rho}{\partial t^2} = \frac{\partial}{\partial x_i} \left(\frac{\partial p}{\partial x_i} + \frac{\partial \rho u_i}{\partial t} \right) = \frac{\partial}{\partial x_i} \left(- \frac{\partial \rho u_j u_i}{\partial x_j} \right).$$

Now applying the ideal gas law $p = \rho R T$ where we approximate the temperature as nearly constant, $T \approx T_0$, and thus the speed of sound may also be approximated as constant, $c = \sqrt{RT} \approx \sqrt{RT_0} \triangleq c_0$, we obtain the inviscid approximation of **Lighthill's equation**,

$$\left(\frac{\partial^2}{\partial x_i^2} - \frac{1}{c_0^2} \frac{\partial^2}{\partial t^2} \right) p = - \frac{\partial^2 T_{ij}}{\partial x_i \partial x_j} \quad \text{where} \quad T_{ij} = \rho u_i u_j. \quad (13.15a)$$

It is convenient that, for low Mach number flows, incompressible simulations may be used to approximate the noise generated by the flow. This may be accomplished by calculating the **acoustic pressure field** radiated according to the wave equation given by (13.15a), with the **Lighthill stress tensor** T_{ij} on the RHS taken as a source term using the **hydrodynamic velocity field** well approximated by the incompressible flow simulation. The boundary conditions at solid surfaces on this system for the acoustic pressure field are derived in a similar manner, with

$$\left(\frac{\partial^2}{\partial x_i^2} - \frac{1}{c_0^2} \frac{\partial^2}{\partial t^2} \right) p = b \quad \text{where} \quad b = \frac{\partial^2 p}{\partial n^2}, \quad (13.15b)$$

where, again, the RHS term b is taken as a source term using the hydrodynamic pressure field well approximated by the incompressible flow simulation.

Note that, by writing the PDE (13.15a) as a first order system

$$\mathbf{q}' = \begin{pmatrix} 0 & 1 \\ c_0^2 \frac{\partial^2}{\partial x_i^2} & 0 \end{pmatrix} \mathbf{q} + \begin{pmatrix} 0 \\ c_0^2 \frac{\partial^2 T_{ij}}{\partial x_i \partial x_j} \end{pmatrix} \quad \text{where} \quad \mathbf{q} = \begin{pmatrix} p \\ p' \end{pmatrix},$$

discretizing in space, and enforcing the boundary conditions (13.15b), this system may easily be solved using the techniques already presented (e.g., Crank-Nicholson in time and Fourier/Padé in space). Approaches based on **Green's functions** may also be employed. ?

13.A Diablo

The numerical algorithm described in the first half of this chapter is implemented in the open-source code **diablo**, available at <http://numerical-renaissance.com/diablo>. Note that the diablo code is distributed under the **GNU General Public License**:

Diablo is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This code is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License (distributed with diablo) for more details.

There are several README files that accompany the code to help get the reader started running it. For clarity, some of the details of the numerical implementation of the spatial and temporal discretization schemes described above are now described further, in order of appearance in the code. Consistent with the notation used above, the RHS of the momentum equations are accumulated in the arrays \mathbf{R}_i . The Runge-Kutta terms will be stored in \mathbf{F}_i (since they are relatively expensive to calculate) and saved for use at the next RK substep. An extra storage array called \mathbf{S} is also defined.

Care has been taken in the numerical implementations in order to minimize both the number of FFTs per timestep and the number of full-sized storage arrays. The algorithms detailed in the following two subsections use 11 full-sized storage arrays. In these presentations, $\hat{\mathbf{u}}_i$, $\hat{\mathbf{p}}$, $\hat{\mathbf{R}}_i$, $\hat{\mathbf{F}}_i$, and $\hat{\mathbf{S}}_i$ denote the Fourier-space representations of \mathbf{u}_i , \mathbf{p} , \mathbf{R}_i , \mathbf{F}_i , and \mathbf{S} . Note that the physical- and Fourier-space representations of any given array occupy the same location in memory, with the FFT transforming from one representation to the other performed in place in the computer memory.

As in the previous section, in physical space, the notation $\mathbf{u}_1 * \mathbf{u}_2$ denotes the pointwise product of the vector \mathbf{u}_1 with the vector \mathbf{u}_2 at each gridpoint. In Fourier space, $\mathbf{k}_x * \hat{\mathbf{u}}_1$ denotes the multiplication of each Fourier coefficient of \mathbf{u}_1 by the corresponding streamwise wavenumber¹², $\hat{\mathbf{R}}_i / \mathbf{k}_x^2$ denotes the division of each Fourier coefficient of \mathbf{u}_1 by the square of the corresponding streamwise wavenumber, etc.

Exercises

References

Pozrikidis, C (1997) *Introduction to Theoretical and Computational Fluid Dynamics*. Oxford.

¹²Note that this is done without ambiguity even though, \mathbf{k}_x is a one-dimensional array of wavenumbers, whereas \mathbf{u}_1 is a three-dimensional array of Fourier coefficients.