

Exact expressions for the unresolved stress in a finite-volume based large-eddy simulation

Syver Døving Agdestein ^{a,b,*}, Roel Verstappen ^c, Benjamin Sanderse ^{a,b}

^a Scientific Computing Group, Centrum Wiskunde & Informatica, Science Park 123, Amsterdam, 1098 XG, The Netherlands

^b Centre for Analysis, Scientific Computing and Applications, Eindhoven University of Technology, PO Box 513, Eindhoven, 5600 MB, The Netherlands

^c Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen, Nijenborgh 9, Groningen, 9747 AG, The Netherlands

ARTICLE INFO

Keywords:

Commutator errors
Closure modeling
Data-consistency
Filtering
Finite volume method
Large-eddy simulation
Sub-filter stress
Turbulence.

ABSTRACT

In this article we propose new discretization-informed expressions for the residual stress tensor (RST) in a finite-volume based large-eddy simulation (LES-FVM). In addition to the classical RST $\overline{u'u'} - \overline{u}u'$ resulting from the non-commutation between filtering and the nonlinear stress, our RST also contains contributions from the numerical flux, discrete divergence, and pressure terms. Unlike the classical RST, our proposed RST is non-symmetric and non-local. The proposed form of the RST is important for generating appropriate reference data for LES closure modeling. Based on DNS results of the 1D Burgers and 3D incompressible Navier-Stokes equations, we show that the discretization-induced parts of the RST play an important role in the LES-FVM equation for common LES filter widths. When the discrete contribution is included, our RST expression gives zero a-posteriori error in LES, while existing RST expressions give errors that increase over time. For a Smagorinsky model, we show that the Smagorinsky coefficient is higher when fitted to our new RST than when fitted to the classical RST and gives improved results.

1. Introduction

Turbulent fluid flows can be modeled by the incompressible Navier-Stokes equations, but they are in general computationally too expensive to solve using direct numerical simulation (DNS). Large eddy simulation (LES) consists of finding equations for the large-scale features of the flow, which are extracted using a spatial filter. The LES equations can be solved using fewer numerical computations.

The incompressible Navier-Stokes equations can be discretized using the finite volume method (FVM). Like LES, the FVM considers filtered velocities, with the filter being the average over a control volume. In LES, this filter can have other definitions, for example a convolution with an arbitrary kernel function. Unlike LES, the FVM equations are discrete by design.

The continuous LES equations include a continuous divergence of a flux function. The discrete FVM equations contain an integral of a flux function over the control volume boundary. By defining this boundary integral as a *discrete* divergence operator applied to the given flux, the FVM equations take the same form as the continuous LES equation. This allows for treating LES and the FVM as the same problem, as suggested by Verstappen [1].

* Corresponding author.

E-mail addresses: sda@cwj.nl (S.D. Agdestein), r.w.c.p.verstappen@rug.nl (R. Verstappen), b.sanderse@cwj.nl (B. Sanderse).

Both the FVM and LES result in approximate equations for the filtered velocity field \bar{u} that are different from the exact filtered conservation law. The mismatch between the model equations and the exact filtered conservation law can be written as a residual term, sometimes called a commutator error.

Most works on the closure problem focus on modeling the commutator between filtering and nonlinearities [2–4]. Assuming filtering commutes with spatial and temporal differentiation, the residual term takes the form of the divergence of a tensor, $\nabla \cdot \tau(u)$, which is a function of the resolved and unresolved velocity fields \bar{u} and u through the residual stress tensor (RST) $\tau_{ij}(u) := \overline{u_i u_j} - \bar{u}_i \bar{u}_j$. The exact LES equations are approximated by replacing $\tau(u)$ with a closure model $m(\bar{u})$. *Functional* models try to match the dissipation produced by $\tau(u)$, while *structural* models also try to model the structure of the RST $\tau(u)$ itself [4,5].

Apart from the difficulty of correctly modeling the residual $\nabla \cdot \tau(u)$, another major challenge in LES is the appearance of additional residual terms when the equations are discretized [6,7]. These terms are commonly referred to as *discretization errors*. For common discretization schemes, they are controlled by convergence bounds on the grid spacing h .

For typical LES scenarios, the filter width Δ is of the same order of magnitude as the grid spacing h [8,9], sometimes with an exact equality $\Delta = h$ [10–13]. For this reason the discretization errors can be of the same order of magnitude as $\nabla \cdot \tau(u)$ [14], which can strongly limit the usefulness of any closure modeling effort that only accounts for $\nabla \cdot \tau(u)$. This was illustrated by Bae and Lozano-Duran, who used DNS to assess the importance of the discretization error by computing the residual $\nabla \cdot \tau(u)$ explicitly [6]. A recognition of this issue has motivated the development of LES frameworks where the total residual from both non-linearities and discretization are modeled [7,11,15]. However, the exact form of this total residual term has not yet been well-defined.

We are interested in the exact discretization-informed expression for the total residual because the final goal is to use such an expression as target data for tuning the parameters of a closure model. In previous work, we showed that using a discretization-informed residual as target data for data-driven closure models gives stable models, while using incorrect target data leads to instabilities [16]. In this work, we introduce a new mathematical formalism to derive exact expressions for the residual appearing in the discretized LES equations. Furthermore, we write the residual in the form of a discrete divergence of an RST. We show how this RST differs between classical LES, the classical FVM, and propose a new discrete LES-FVM framework that merges the two.

The main ideas in this article are visualized in Fig. 1. The notation used in the figure will be explained in detail in the following sections. All equations are derived from the original conservation law for the turbulent solution u in Fig. 1-A. The approach we use is to first transform the equation for u by applying LES and FVM filters. The equations are then rewritten into the desired form, and the residual terms are gathered in an unresolved stress. The approximation steps (called “closures”) are only performed after the exact residual stress has been defined. These approximation steps are indicated by a change of color in the figure.

In Section 2, we first introduce our notation. For 1D conservation laws, we derive classical LES and the FVM in this notation, so that in Section 3, we can merge classical LES and the FVM into a unified discrete LES-FVM framework. We derive new exact discretization-informed RST expressions for the given framework. In Section 4, the importance of the RST definition is tested for the 1D Burgers equation. We show that using our discretization-informed RST as a closure term gives a “perfect” closure model with zero a-posteriori error, while the classical RST gives errors that accumulate over time. This difference remains visible when the RSTs are used as target data in a Smagorinsky model. In Section 5, we extend our LES-FVM framework to the 3D incompressible Navier-Stokes equations. Due to the incompressibility, the RST is shown to be non-local. The RST is also shown to be non-symmetric. In Section 6, we repeat the Burgers experiments for a 3D decaying turbulence simulation, confirming that our expressions are correct for the full incompressible Navier-Stokes equations.

2. Preliminaries: LES and the FVM

Let $\Omega = [0, \ell]$ be a 1D domain with length $\ell > 0$. Let $U := \{u : \mathbb{R} \rightarrow \mathbb{R} \mid u(x) = u(x + \ell)\}$ be the space of periodic functions on Ω . Depending on the problem, the space U may need to be further restricted to more regular spaces such as $L^2(\Omega)$ or $H^1(\Omega)$ [2].

Consider the generic infinitesimal 1D conservation law

$$L(u) := \partial_t u + \partial_x r(u) = 0 \quad (1)$$

(see Fig. 1-A), where $u(x, t)$ is the solution at a given point x and time t , $L := \partial_t + \partial_x r$ is the equation operator, $\partial_t := \partial/\partial t$ and $\partial_x := \partial/\partial x$ are partial derivatives, and the flux $r : U \rightarrow U$ is a non-linear spatial operator. We call the conservation law *infinitesimal* since the divergence ∂_x and potentially the flux r are infinitesimal operators (as opposed to discrete operators that can be computed on a grid). For simplicity, we did not include a source term in (1). Adding a source term is straightforward and does not change the derivations in this article.

The conservation law (1) is a continuous equation defined by requiring that the field $L(u) \in U$ is zero everywhere. We can evaluate $L(u)$ in a point $x \in \Omega$ and time $t \geq 0$ as $L(u)(x, t) \in \mathbb{R}$. In the following, we omit the time t and write $L(u)(x) \in \mathbb{R}$ and $u(x) \in \mathbb{R}$ etc.

An example of a non-linear conservation law is the viscous Burgers equation. The corresponding flux is defined as

$$r(u) := \frac{1}{2} u u - \nu \partial_x u, \quad (2)$$

where $\nu > 0$ is a constant viscosity (diffusion coefficient).

The PDE (1) can be discretized and solved directly by using the FVM. Alternatively, the equation can first be filtered and approximated with an LES closure. To distinguish between these two approaches, we use the superscript $(\cdot)^\Delta$ for quantities related to LES, and $(\cdot)^h$ for quantities related to the FVM.

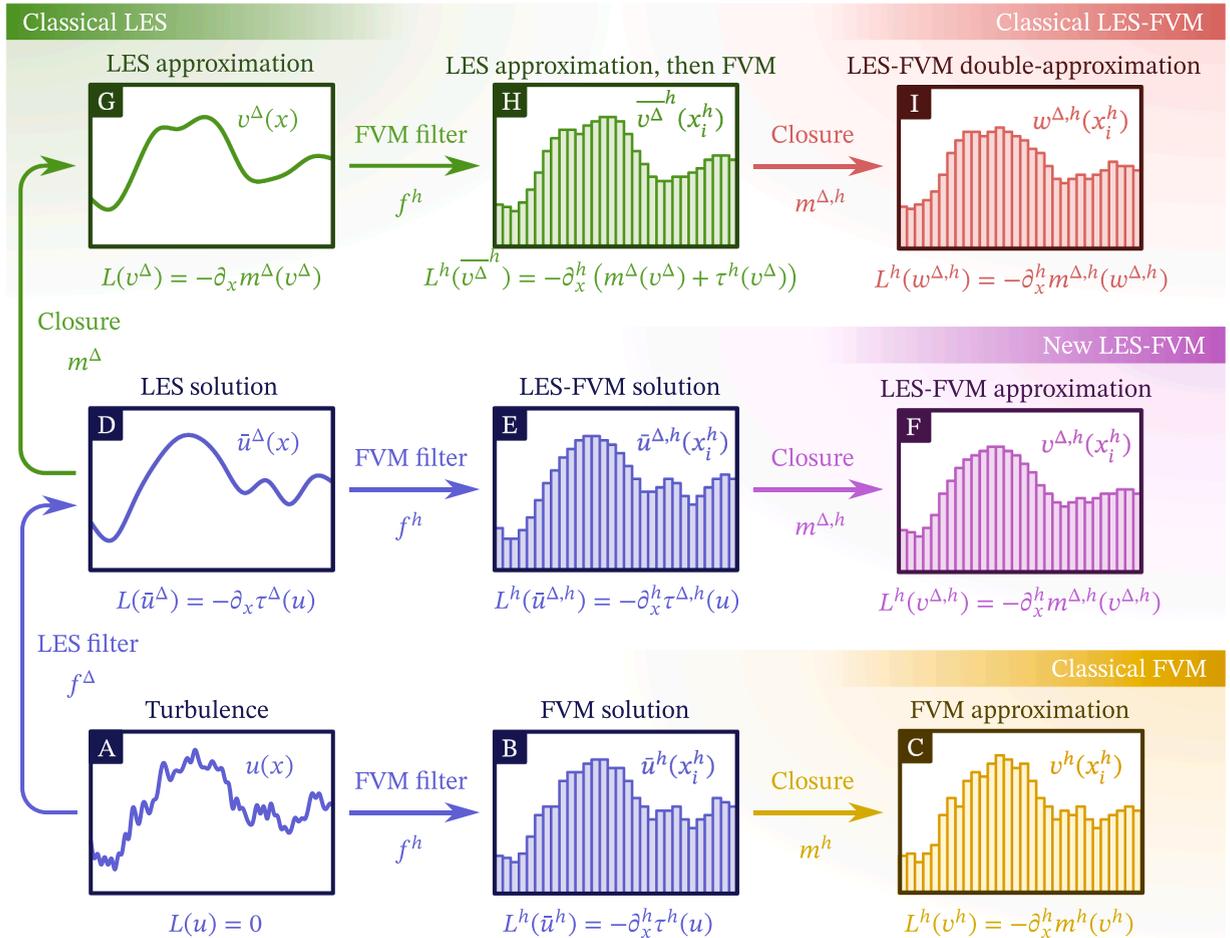


Fig. 1. Three different routes to a closed system of discrete equations that simulate the large-scales of a turbulent flow u . A change of color indicates an approximation step, while a preservation of color indicates an exact operation. The grid points are $(x_i^h)_{i \in \mathbb{Z}}$. Classical LES-FVM relies on two separate approximation steps, whereas the FVM and our proposed LES-FVM framework only rely on one approximation step.

2.1. Filtering and LES

Eq. (1) describes all the scales of motion for the given system. Filtering (1) with a convolutional LES filter

$$f^\Delta : U \rightarrow U, u \mapsto \bar{u}^\Delta \tag{3}$$

of filter-width $\Delta \geq 0$ gives the LES equation

$$\overline{L(u)}^\Delta = 0, \tag{4}$$

where $\bar{u}^\Delta := f^\Delta u$ is the LES solution that we intend to solve for.

The convolution f^Δ is defined through a kernel $G^\Delta : \mathbb{R} \rightarrow \mathbb{R}$ by

$$\bar{u}^\Delta(x) := \int_{\mathbb{R}} G^\Delta(x-y)u(y) dy \tag{5}$$

for all $u \in U$ and $x \in \Omega$. Note that we integrate over \mathbb{R} (not Ω), since the filter kernel G^Δ needs to be extended beyond the periodic boundary. Some kernels (such as the Gaussian kernel) have infinite support. In practice, such kernels are truncated, and one periodic extension is sufficient.

It is common to decompose the LES equation into a resolved and unresolved part as

$$\begin{aligned} L(\bar{u}^\Delta) &= -(\overline{L(u)}^\Delta - L(\bar{u}^\Delta)) \\ &= -(\overline{\partial_x r(u)}^\Delta - \partial_x r(\bar{u}^\Delta)), \end{aligned} \tag{6}$$

where the left-hand side only depends on the resolved scales \bar{u}^Δ and the right-hand side is a residual that still depends on the full solution u . This term is not yet the “divergence of a flux”, meaning that (6) is not expressed as a conservation law. However, since f^Δ is a convolution, it satisfies the *filter-swap* commutation property

$$f^\Delta \partial_x = \partial_x f^\Delta. \quad (7)$$

For proof, see [Theorem 1](#) in [Appendix C](#). This can also be written as $\overline{\partial_x u}^\Delta = \partial_x \bar{u}^\Delta$ for all $u \in U$. This commutation property can be used to obtain a *conservative* form of the residual and rewrite [Eq. \(6\)](#) as a conservation law

$$L(\bar{u}^\Delta) = -\partial_x \tau^\Delta(u) \quad (8)$$

(see [Fig. 1-D](#)), where the commutator between L and f^Δ takes the form of the divergence of a residual flux

$$\tau^\Delta(u) := \overline{r(u)^\Delta} - r(\bar{u}^\Delta). \quad (9)$$

This flux is also known as the *sub-filter* flux. For the Burgers equation, we get the well-known expression $\tau^\Delta(u) = (\overline{uu}^\Delta - \bar{u}^\Delta \bar{u}^\Delta)/2$. As $\tau^\Delta(u)$ is a flux, it has dissipation properties that could potentially be replicated by a closure model. For example, a convective flux conserves energy, while a diffusive flux dissipates energy. Since $\partial_x \tau^\Delta(u)$ is of conservative form, the filtered momentum is conserved, which cannot be guaranteed otherwise. We highlight these properties and how they are obtained here since they do not automatically apply in the discrete case.

[Eq. \(8\)](#) is exact, but unclosed. The next step in LES is to approximate the residual flux by a closure model $m^\Delta(\bar{u}^\Delta) \approx \tau^\Delta(u)$.¹ Closure models are often chosen in the functional eddy-viscosity form:

$$m^\Delta(\bar{u}^\Delta) := -v^\Delta(\bar{u}^\Delta) \partial_x \bar{u}^\Delta, \quad (10)$$

where $v^\Delta : U \rightarrow U$ is an eddy viscosity chosen such that the dissipation produced by $m^\Delta(\bar{u}^\Delta)$ is similar to the one produced by $\tau^\Delta(u)$. For the basic Smagorinsky model [\[17\]](#), the viscosity is

$$v^\Delta(\bar{u}^\Delta) := \theta^2 \Delta^2 |\partial_x \bar{u}^\Delta|, \quad (11)$$

where $\theta > 0$ is a model parameter.

The approximate LES equation is

$$L(v^\Delta) = -\partial_x m^\Delta(v^\Delta) \quad (12)$$

(see [Fig. 1-G](#)), where $v^\Delta \approx \bar{u}^\Delta$ is the approximate LES solution. It is in general different from \bar{u}^Δ since the closure m^Δ cannot be exact when information is lost in the filtering process.

To summarize, we used the following steps to derive a closed form for the approximate equation in LES:

1. Apply the LES filter f^Δ .
2. Decompose the equation into a resolved part and a residual.
3. Rewrite the residual in conservative form using the filter-swap property.
4. Approximate the residual flux with a closure model.

These steps are visualized in the route A→D→G in [Fig. 1](#).

[Eq. \(12\)](#) still needs to be discretized. The discretization we use is the FVM. As we will see, the equations for the FVM can be derived in a similar way as for LES.

2.2. The FVM through filter-swap

Let $h > 0$ denote the grid spacing of a uniform grid on Ω . The derivations in this section hold for all points $x \in \Omega$, and we do not restrict any quantities to the grid points until [Section 4](#). For now, h can be considered as a particular length scale that is independent from the LES filter width Δ from [Section 2.1](#).

2.2.1. Numerical derivatives and fluxes

For all $u \in U$ and $x \in \Omega$, define the staggered central finite difference and interpolation operators as

$$\partial_x^h u(x) := \frac{1}{h} \left[u\left(x + \frac{h}{2}\right) - u\left(x - \frac{h}{2}\right) \right], \quad (13)$$

$$\eta_x^h u(x) := \frac{1}{2} \left[u\left(x - \frac{h}{2}\right) + u\left(x + \frac{h}{2}\right) \right]. \quad (14)$$

¹ We use the symbol “ \approx ” solely to indicate the *intent* of an approximation to guide the reader. The statement “ $a \approx b$ ” provides no guarantee that a is in any way similar or close to b . Instead, we take care to use different symbols for new quantities arising from an approximation step, for example $v^\Delta \neq \bar{u}^\Delta$.

These operators can be used to discretize the conservation law (1). If $r^h \approx r$ is a numerical flux, then $\partial_x^h r^h \approx \partial_x r$ is an approximation of the flux term, and $L^h := \partial_t + \partial_x^h r^h$ is a (spatial) approximation of the conservation law. For the Burgers equation (defined by Eq. (2)), we use the second-order accurate numerical flux

$$r^h(u) := \frac{1}{2} (\eta_x^h u) (\eta_x^h u) - \nu \partial_x^h u. \tag{15}$$

The operators $\partial_x^h : U \rightarrow U$, $\eta_x^h : U \rightarrow U$, and $r^h : U \rightarrow U$ are both continuous and discrete at the same time. They are continuous in the sense that $\partial_x^h u(x)$ can be evaluated for all $x \in \Omega$. They are discrete in the sense that $\partial_x^h u(x)$ only depends on $u(x \pm h/2)$, and for the numerical Burgers flux (15), $\partial_x^h r^h(u)(x)$ only depends on $u(x - h)$, $u(x)$, and $u(x + h)$. We therefore say that these operators are *grid-compatible* or *h-compatible*. See Appendix A for more details about infinitesimal, discrete, continuous, and grid-compatible operators.

2.2.2. The FVM grid filter

The finite difference ∂_x^h is closely related to the FVM grid filter

$$f^h : U \rightarrow U, u \mapsto \bar{u}^h \tag{16}$$

defined for all $x \in \Omega$ as

$$\bar{u}^h(x) := \frac{1}{h} \int_{x-h/2}^{x+h/2} u(y) dy. \tag{17}$$

This filter is sometimes called a “top-hat filter”, “Schumann’s filter” [18], or “volume-averaging filter”, since it averages u over a control volume $[x \pm h/2]$. The FVM filter f^h constitutes a particular choice of convolutional LES filter f^Δ , where the filter width Δ is equal to the grid spacing h used in the finite difference ∂_x^h . The underlying top-hat kernel G^h is

$$G^h(x) := \begin{cases} \frac{1}{h} & \text{if } |x| \leq \frac{h}{2}, \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

A filter width equal to the grid spacing is commonly the setting in implicit LES, but we emphasize that here the FVM filter definition is known explicitly.

The infinitesimal operator ∂_x is not grid-compatible. By applying the FVM filter f^h , the derivative ∂_x can be turned into the grid-compatible finite difference ∂_x^h through the discrete filter-swap property

$$\partial_x^h = f^h \partial_x, \tag{19}$$

which can also be written as $\partial_x^h u = \overline{\partial_x u}^h$ for all $u \in U$. The proof is given in Theorem 3. This property entails that the finite difference ∂_x^h is equal to a filtered version of the exact derivative ∂_x , as noted by Schumann and others [18–20]. If we approximate an infinitesimal derivative ∂_x by ∂_x^h , the content of the derivative is implicitly filtered. The FVM filter f^h is therefore sometimes called a “discretization-induced filter” or an “implicit filter” induced by the choice of ∂_x^h . We prefer to make this statement explicit by turning it around and explicitly applying f^h to a derivative ∂_x . The finite difference ∂_x^h is therefore a “filter-induced discretization” of the infinitesimal derivative ∂_x .

We see the property (19) as an analogous version of the commutation property (7) for the finite difference operator ∂_x^h and grid filter f^h . The subtle, yet crucial, difference with (7) is that the left-hand side is no longer filtered and uses a discrete differentiation operator. An intuitive way to understand this property is through the use of a telescoping sum. See Fig. C.15 in Appendix C for a visualization of how the inner terms in the integral defining f^h cancel out in the discrete filter-swap property. Unlike the property (7), we have $\partial_x^h f^h \neq f^h \partial_x$ (see Theorem 2 for proof).

2.2.3. FVM Equations

To obtain FVM equations, we use the same approach as we used for LES in Section 2.1:

1. Apply the FVM filter f^h .
2. Decompose the equation into a resolved part and a residual.
3. Rewrite the residual in conservative form using the discrete filter-swap property.
4. Approximate the residual flux with a closure model.

These steps are visualized in the bottom route in Fig. 1.

Applying the FVM filter f^h to Eq. (1) yields the FVM equation

$$\overline{L(u)}^h = 0, \tag{20}$$

where \bar{u}^h is the FVM solution that we intend to solve for. While the classical LES formulation could be obtained by using the filter-swap property $f^h \partial_x = \partial_x f^h$, our aim is to have an equation for \bar{u}^h that involves the discrete divergence ∂_x^h . This is achieved by applying the discrete filter-swap property to Eq. (20):

$$\partial_t \bar{u}^h + \partial_x^h r(u) = 0. \tag{21}$$

This equation is more commonly written as

$$h\partial_t \bar{u}^h(x) + r(u)\left(x + \frac{h}{2}\right) - r(u)\left(x - \frac{h}{2}\right) = 0, \quad (22)$$

emphasizing that we have two unknown fluxes at the left and right boundaries of the control volume $[x \pm h/2]$.

Eq. (22) is often derived from the integral form of the conservation law (1):

$$\frac{d}{dt} \int_V u dV + \int_{\partial V} r(u) \cdot n dS = 0, \quad \forall V \subset \Omega. \quad (23)$$

In 1D, with $V = [x \pm h/2]$, $\partial V = \{x \pm h/2\}$, and $n = \pm 1$, this is exactly Eq. (22).

Let $r^h : U \rightarrow U$ denote a grid-compatible numerical flux that approximates the original flux r . By adding the resolved term $\partial_x^h r^h(\bar{u}^h)$ to both sides of Eq. (21) and rearranging the terms, we get

$$L^h(\bar{u}^h) = -\partial_x^h \tau^h(u) \quad (24)$$

(see Fig. 1-B), where $L^h := \partial_t + \partial_x^h r^h$ is a grid-compatible approximation of the original conservation law operator L and

$$\tau^h(u) := r(u) - r^h(\bar{u}^h) \quad (25)$$

is the residual flux in the FVM equation. Unlike the LES Eq. (8), the FVM Eq. (24) is “ ∂_x^h -conservative”, since the conservation law is written with respect to the discrete divergence ∂_x^h . The residual flux $\tau^h(u)$ is also known as the *sub-grid* flux. Calling it a sub-filter flux would be misleading, since it also depends on the choice of numerical divergence ∂_x^h and numerical flux r^h (in addition to the FVM filter f^h). We use the term *residual flux* for both $\tau^\Delta(u)$ and $\tau^h(u)$.

By approximating the residual flux with a grid-compatible dissipation model $m^h(\bar{u}^h) \approx \tau^h(u)$, we get the approximate FVM equation

$$L^h(v^h) = -\partial_x^h m^h(v^h) \quad (26)$$

(see Fig. 1-C), where $v^h \approx \bar{u}^h$ is the approximate FVM solution. In a properly resolved DNS, where h is sufficiently small, the dissipation model m^h can be set to zero.

The operators L^h and m^h are still continuous and can be evaluated in every point $x \in \Omega$. “Fully discrete” equations for v^h can be obtained by simply evaluating Eq. (26) at the grid points $x_i^h := ih$, $i \in \{1, \dots, N\}$ on the condition that $h = \ell/N$ for some integer $N \in \mathbb{N}$. This restriction is visualized in Fig. 1-B and Fig. 1-C. The grid-compatible numerical fluxes r^h and m^h ensure that the restricted equations form a closed system of ordinary differential equations.

While Eq. (24) can be seen as a discrete LES equation, the filter width is implied by the grid spacing h used in the discrete divergence ∂_x^h . In LES, we would like to be able to choose the filter width Δ independently from h . We therefore propose to combine the LES filter f^Δ with the FVM filter f^h .

3. Combining LES and the FVM: LES-FVM

Classical LES and the FVM both aspire to correctly model the features of the flow that are larger than a certain length scale. In LES, the large scales are extracted using an LES filter f^Δ , which retains the scales that are larger than the filter width Δ . In the FVM, the size of the resolved scales are inherently linked to the grid size h through the FVM filter f^h , numerical flux r^h , and discrete divergence ∂_x^h . The fluxes in the LES Eq. (8) are defined with respect to the infinitesimal divergence ∂_x , but the fluxes in FVM Eq. (24) are defined with respect to the discrete divergence ∂_x^h .

In this section, we present a new discrete LES formalism—“LES-FVM”—that bridges the gap between infinitesimal LES and the discrete FVM. In LES-FVM, the LES filter is applied to selectively extract the scales of interest, while the FVM filter is applied to make the divergence grid-compatible. We first show the classical approach in which one approximates before discretizing (top route in Fig. 1), and then we propose our new approach in which we discretize first (middle route in Fig. 1).

3.1. The classical approach: LES first, then FVM

After closure, the approximate LES equations are $L(v^\Delta) = -\partial_x m^\Delta(v^\Delta)$ (see Fig. 1-G). Applying the FVM filter f^h , filter-swap property, adding the resolved term $\partial_x^h r^h(\overline{v^\Delta}^h)$, and rearranging the terms gives

$$L^h\left(\overline{v^\Delta}^h\right) = -\partial_x^h\left(m^\Delta(v^\Delta) + \tau^h(v^\Delta)\right) \quad (27)$$

(see Fig. 1-H), where $\overline{v^\Delta}^h$ is the FVM average of the LES approximation v^Δ . This equation is unclosed in terms of $\overline{v^\Delta}^h$ since it contains the field v^Δ in the right-hand side flux $m^\Delta(v^\Delta) + \tau^h(v^\Delta)$. Note that $\tau^h(v^\Delta)$ is the residual FVM flux from Eq. (25) applied to the LES approximation v^Δ instead of u .

By using a grid-compatible closure model $m^{\Delta,h}(\overline{v^\Delta}^h) \approx m^\Delta(v^\Delta) + \tau^h(v^\Delta)$, we get the approximate equation

$$L^h(w^{\Delta,h}) = -\partial_x^h m^{\Delta,h}(w^{\Delta,h}) \quad (28)$$

(see Fig. 1-I), where $w^{\Delta,h} \approx \overline{v^\Delta}^h$ is the approximate solution. We call $m^{\Delta,h}$ a *closure* since it approximates an unresolved term and thus makes the equation closed. Typically, $m^{\Delta,h}$ is just a “discretization” of the infinitesimal LES closure m^Δ , but the expression

$m^\Delta(v^\Delta) + \tau^h(v^\Delta)$ suggests that $m^{\Delta,h}$ should also account for the FVM discretization error $\tau^h(v^\Delta)$. This classical approach involves multiple steps: close first, then discretize, and then adjust the closure to account for the discretization (as visualized in the top route of Fig. 1).

We now show what this classical approach would entail for the basic Smagorinsky closure model. The LES closure

$$m^\Delta(\bar{u}^\Delta) := -\theta^2 \Delta^2 \left| \partial_x \bar{u}^\Delta \right| \partial_x \bar{u}^\Delta \quad (29)$$

is used for the LES flux $\tau^\Delta(u)$, while its FVM variant

$$m^h(\bar{u}^h) := -\theta^2 h^2 \left| \partial_x^h \bar{u}^h \right| \partial_x \bar{u}^h, \quad (30)$$

is used for the FVM flux $\tau^h(u)$. A natural choice for $m^{\Delta,h}$ is then

$$m^{\Delta,h}(\bar{v}^{\Delta,h}) := -\theta^2 (\Delta^2 + h^2) \left| \partial_x^h \bar{v}^{\Delta,h} \right| \partial_x^h \bar{v}^{\Delta,h}. \quad (31)$$

The term $\theta^2 \Delta^2 \left| \partial_x^h \cdot \right| \partial_x^h \cdot$ is a discretization of m^Δ , while $\theta^2 h^2 \left| \partial_x^h \cdot \right| \partial_x^h \cdot$ is a discrete Smagorinsky model for the residual FVM flux τ^h . The final model $m^{\Delta,h}$ takes the form of a discrete Smagorinsky model for a filter of width $\sqrt{\Delta^2 + h^2}$. The incorporation of h into the filter width can be seen as an adjustment to account for discretization error $\tau^h(v^\Delta)$.

Interestingly, if the LES filter f^Δ is a Gaussian or top-hat filter, then $\sqrt{\Delta^2 + h^2}$ is the width of the LES-FVM *double filter*²

$$f^{\Delta,h} := f^h f^\Delta \quad (32)$$

composed of the LES filter f^Δ and the FVM filter f^h . The field $w^{\Delta,h}$ can therefore be seen as a Smagorinsky approximation of the LES-FVM solution $\bar{u}^{\Delta,h} := f^{\Delta,h} u$.

In the next section, we propose to approximate $\bar{u}^{\Delta,h}$ directly, using one single approximation step $v^{\Delta,h} \approx \bar{u}^{\Delta,h}$, instead of performing successive approximations $v^\Delta \approx \bar{u}^\Delta$ and $w^{\Delta,h} \approx \bar{v}^{\Delta,h}$.

3.2. Our new approach: LES-FVM

We discretize the infinitesimal LES equation $\overline{L(u)}^\Delta = 0$ by applying the FVM filter f^h . This gives the double filtered conservation law

$$\overline{L(u)}^{\Delta,h} = 0. \quad (33)$$

By using the filter-swap property, adding the resolved term $\partial_x^h r^h(\bar{u}^{\Delta,h})$ to both sides, and rearranging the terms, we get an equation for $\bar{u}^{\Delta,h}$ in ∂_x^h -conservative form:

$$L^h(\bar{u}^{\Delta,h}) = -\partial_x^h \tau^{\Delta,h}(u) \quad (34)$$

(see Fig. 1-E), where

$$\tau^{\Delta,h}(u) := \overline{r(u)}^\Delta - r^h(\bar{u}^{\Delta,h}) \quad (35)$$

is the residual flux in the LES-FVM equation. This is a novel residual flux expression that accounts for the errors in both LES and the FVM. The first term only uses the LES filter f^Δ and the infinitesimal flux r , while the second term uses the LES-FVM filter $f^{\Delta,h}$ and the numerical flux r^h .

Using a discretization-informed grid-compatible closure model $m^{\Delta,h}(\bar{u}^{\Delta,h}) \approx \tau^{\Delta,h}(u)$ gives the approximate LES-FVM equation

$$L^h(v^{\Delta,h}) = -\partial_x^h m^{\Delta,h}(v^{\Delta,h}) \quad (36)$$

(see Fig. 1-F), where $v^{\Delta,h} \approx \bar{u}^{\Delta,h}$ is the approximate LES-FVM solution. Although the equation for $v^{\Delta,h}$, obtained by “discretizing first, then approximating”, has the same form as the equation for $w^{\Delta,h}$ in (28), obtained through multiple alternating filtering and approximation steps, it leads to a different solution because the closure is chosen differently. In Eq. (28), $m^{\Delta,h}$ is chosen such that $w^{\Delta,h} \approx \bar{v}^{\Delta,h}$, which depends on a previous approximation $v^\Delta \approx \bar{u}^\Delta$. In Eq. (36), $m^{\Delta,h}$ is chosen such that $v^{\Delta,h} \approx \bar{u}^{\Delta,h}$ directly.

Having defined our new LES-FVM framework, we now proceed to discuss its implications.

3.3. Control over the LES and FVM filter widths

For $f^{\Delta,h}$, we are free to choose Δ and h independently. Choosing $\Delta \gg h$ would result in a grid that resolves scales that are not present in the LES solution \bar{u}^Δ , leading to unnecessary computational costs. This setting is sometimes referred to as *explicit LES*, where the effects of the discretization can be neglected. The advantage is that the classical LES framework can be used with small discretization errors. In this case, we have $\tau^{\Delta,h}(u) \approx \tau^\Delta(u)$. On the other hand, if $\Delta = 0$, we have $f^{\Delta,h} = f^h$, and we recover the classical

² Note that double filters in LES are also used for dynamically determining closure model coefficients by applying a test-filter on top of the LES filter [21]. Our double filter serves a different purpose: to filter out scales smaller than Δ while also coarse-graining the differential operators so that they become compatible with a grid of spacing h .

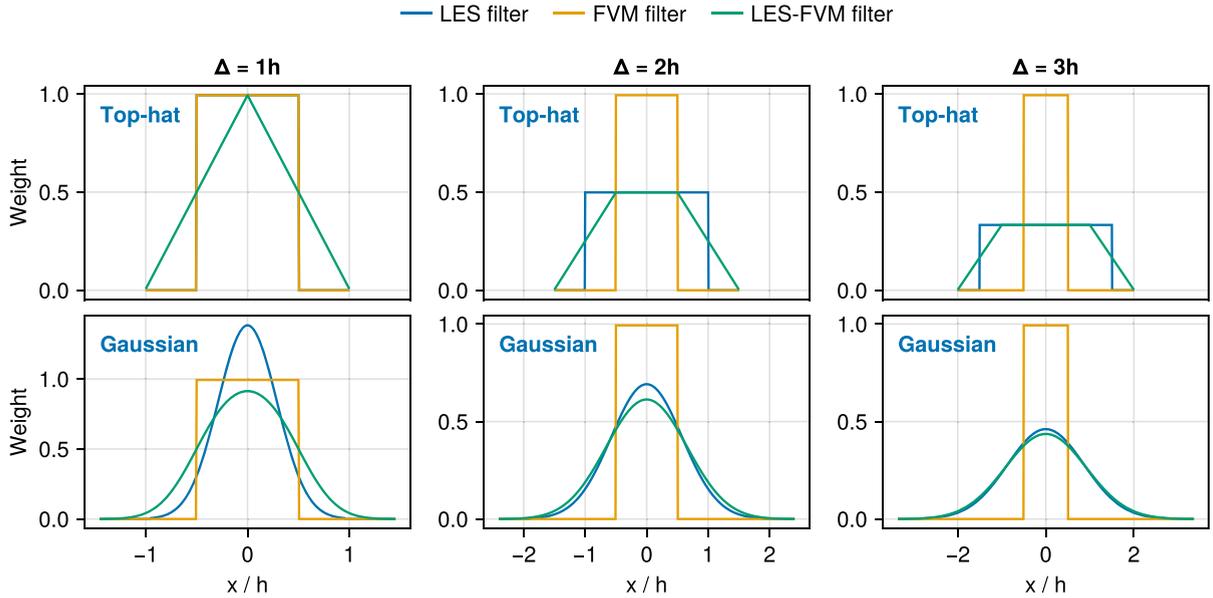


Fig. 2. Spatial kernels of LES filter f^Δ (top-hat and Gaussian), FVM grid filter f^h (always top-hat), and LES-FVM double filter $f^{\Delta,h}$ for $\Delta \in \{1h, 2h, 3h\}$. In the top-left plot, we have $f^\Delta = f^h$.

FVM setting. This setting is sometimes referred to as *implicit LES*, where no explicit LES filter f^Δ is used. In literature, common choices for the ratio are $0 \leq \Delta/h \leq 4$ [9,11]. In this range, the effect of the FVM filter is equal to or somewhat smaller than the effect of the LES filter. It is generally acknowledged that the ratio Δ/h should ideally be higher to prevent interference from the discretization in the residual (for example $\Delta/h \geq 4$ for a second-order accurate discretization [8] or $\Delta/h \geq 2$ for a sixth-order accurate discretization [22]).

Instead of circumventing the problem of the discretization in LES by choosing $\Delta \gg h$, we aim to allow for smaller ratios Δ/h by explicitly accounting for the discretization effects in our new residual flux $\tau^{\Delta,h}(u)$. In our new LES-FVM framework, all values of the ratio Δ/h are valid choices.

In Fig. 2, we show the filter kernels G^Δ , G^h , and $G^{\Delta,h}$ corresponding to the filters f^Δ , f^h , and $f^{\Delta,h}$ for top-hat and Gaussian LES kernels

$$G_{\text{top-hat}}^\Delta(x) := \begin{cases} \frac{1}{\Delta} & \text{if } |x| \leq \frac{\Delta}{2}, \\ 0 & \text{otherwise} \end{cases}, \tag{37}$$

$$G_{\text{Gaussian}}^\Delta(x) := \sqrt{\frac{6}{\pi\Delta^2}} \exp\left(-\frac{6x^2}{\Delta^2}\right). \tag{38}$$

The LES filter widths are $\Delta \in \{1h, 2h, 3h\}$. For top-hat G^Δ with $\Delta = h$, we have $G^\Delta = G^h$, and $G^{\Delta,h}$ becomes a triangular kernel. For top-hat G^Δ with $\Delta > h$, the double filter kernel $G^{\Delta,h}$ takes the shape of a trapezoid, with $G^{\Delta,h}(x) = G^\Delta(x)$ for $|x| \leq (\Delta - h)/2$. For Gaussian G^Δ , the double filter kernel $G^{\Delta,h}$ resembles a Gaussian kernel of width $\sqrt{\Delta^2 + h^2}$. However, $G^{\Delta,h}$ is not exactly Gaussian. As the ratio Δ/h grows, G^Δ and $G^{\Delta,h}$ become more and more similar. For the ratios shown, G^Δ and $G^{\Delta,h}$ are still visually different.

For the top-hat and Gaussian LES filters, the spectral transfer functions are defined as [3]

$$\hat{G}_{\text{top-hat}}^\Delta(\kappa) := \frac{\sin(\kappa\Delta/2)}{\kappa\Delta/2}, \tag{39}$$

$$\hat{G}_{\text{Gaussian}}^\Delta(\kappa) := \exp\left(-\frac{\kappa^2\Delta^2}{24}\right), \tag{40}$$

where $\kappa := 2\pi k$ for $k \in \mathbb{Z}$. Interestingly, they both give the same Taylor series expansion up to second-order around $\kappa = 0$:

$$\hat{G}_{\text{top-hat}}^\Delta(\kappa) = 1 - \frac{\kappa^2\Delta^2}{24} + \mathcal{O}(\kappa^4), \tag{41}$$

$$\hat{G}_{\text{Gaussian}}^\Delta(\kappa) = 1 - \frac{\kappa^2\Delta^2}{24} + \mathcal{O}(\kappa^4). \tag{42}$$

For the double-filter $f^{\Delta,h}$, we therefore get the expansion

$$\hat{G}^{\Delta,h}(\kappa) = \hat{G}^\Delta(\kappa)\hat{G}^h(\kappa) = 1 - \frac{\kappa^2(\Delta^2 + h^2)}{24} + \mathcal{O}(\kappa^4) \tag{43}$$

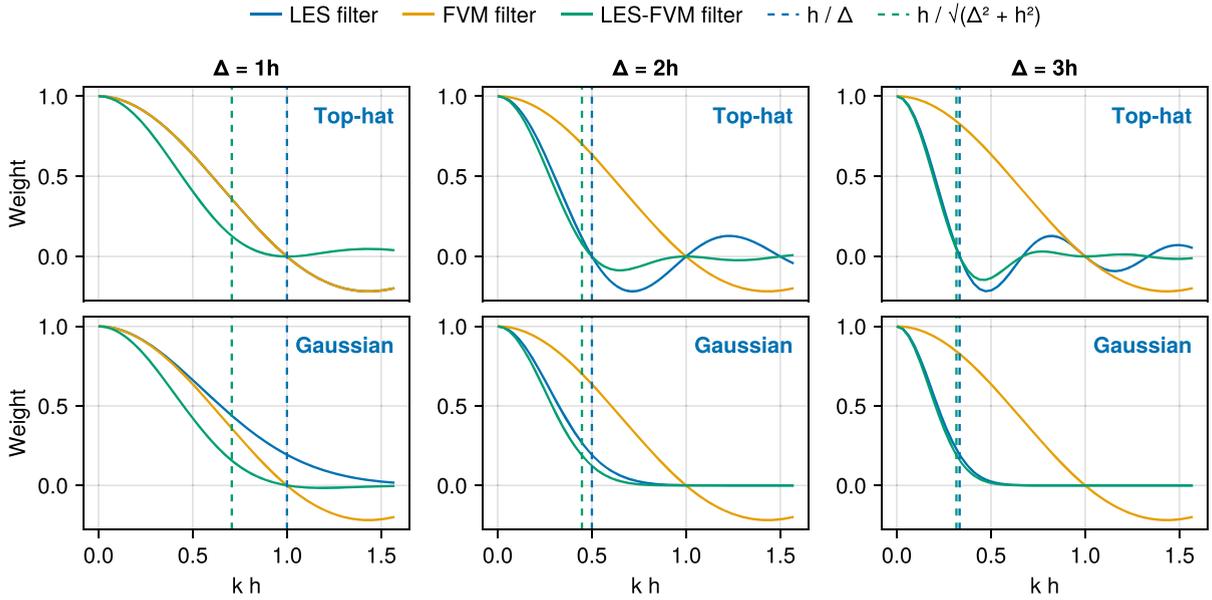


Fig. 3. Spectral transfer functions of LES filter f^Δ (top-hat and Gaussian), FVM grid filter f^h (always top-hat), and LES-FVM double filter $f^{\Delta,h}$ for $\Delta \in \{1h, 2h, 3h\}$. In the top-left plot, we have $f^\Delta = f^h$.

for both the top-hat and Gaussian LES kernels. By comparing the second-order terms, we can make the argument that $\sqrt{\Delta^2 + h^2}$ is indeed the width of the LES-FVM double filter.

The spectral transfer functions \hat{G}^Δ , \hat{G}^h , and $\hat{G}^{\Delta,h}$ are shown in Fig. 3 as a function of the wavenumber $k \in \mathbb{Z}$. As the ratio Δ/h grows, the transfer functions $\hat{G}^\Delta(\kappa)$ and $\hat{G}^{\Delta,h}(\kappa) = \hat{G}^\Delta(\kappa)\hat{G}^h(\kappa)$ become more and more similar. Their respective filter widths (represented by vertical dashed lines) also get closer. For $\Delta = h$, $\hat{G}^{\Delta,h}(\kappa)$ is quite different from $\hat{G}^\Delta(\kappa)$ and $\hat{G}^h(\kappa)$, and dampens at much lower wavenumbers.

3.4. Interpretation in terms of existing residual flux expressions

The residual LES-FVM flux $\tau^{\Delta,h}$ can be decomposed in two ways in order to relate it to the classical residual LES flux τ^Δ . The first decomposition is

$$\tau^{\Delta,h}(u) = \tau^\Delta(u) + \tau^h(\bar{u}^\Delta), \tag{44}$$

meaning that the FVM step induces an additional residual flux $\tau^h(\bar{u}^\Delta)$ in addition to the residual LES flux $\tau^\Delta(u)$. The second decomposition is obtained by writing $\tau^{\Delta,h}$ in terms of the classical residual LES flux for the double filter $f^{\Delta,h}$:

$$\tau^{\Delta,h}(u) = \tau_{\text{classic}}^{\Delta,h}(u) + \tau_{\text{flux}}^{\Delta,h}(u) + \tau_{\text{div}}^{\Delta,h}(u), \tag{45}$$

where

$$\tau_{\text{classic}}^{\Delta,h}(u) := \overline{r(u)}^{\Delta,h} - r(\bar{u}^{\Delta,h}), \tag{46}$$

$$\tau_{\text{flux}}^{\Delta,h}(u) := r(\bar{u}^{\Delta,h}) - r^h(\bar{u}^{\Delta,h}), \tag{47}$$

$$\tau_{\text{div}}^{\Delta,h}(u) := \overline{r(u)}^\Delta - \overline{r(u)}^{\Delta,h} \tag{48}$$

are the classical residual LES flux for the double filter $f^{\Delta,h}$, a discretization error from the numerical flux $r^h \approx r$, and a discretization error from the discrete divergence $\partial_x^h \approx \partial_x$, respectively. Due to the filter-swap property, $\tau_{\text{div}}^{\Delta,h}(u)$ can be written as a *high-pass grid filter* $1 - f^h$ applied to the LES flux $\overline{r(u)}^\Delta$. A similar observation was made by Geurts and van der Bos [11]. Our residual flux $\tau^{\Delta,h}(u)$ still has two important differences from the expression of Geurts and van der Bos. First, our residual flux contains the numerical flux r^h in addition to the exact flux r . Second, we define the residual flux with respect to the *discrete* divergence ∂_x^h (through the filter-swap property), whereas Geurts and van der Bos defined their residual flux with respect to the *infinitesimal* divergence ∂_x (through the reverse filter-swap property).

For the FVM, Winckelmans [23], Denaro [24,25] and Verstappen [1] derived a similar expression for the residual flux as τ^h in Eq. (25), where only one of the two terms involve the filter. In our notation, their residual flux resembles $r(u) - r(\bar{u}^h)$, with r instead of r^h in the resolved term. Our expression τ^h is different in the sense that it also accounts for the numerical flux r^h . For the Burgers flux (15), r^h includes the interpolation and finite difference operators η_x^h and ∂_x^h . Verstappen instead considered the interpolation η_x^h explicitly, as a second filter. He analyzed the filtered equations in terms of the “1h-filter” f^h and the “2h-filter” $\eta_x^h f^h$ (their filter

widths are h and $2h$ respectively). This leads to an orthogonal decomposition of the energy spectrum into three parts: a resolved part, a sub- $2h$ -filter part, and a sub- h -filter part.

3.5. Can we discretize without the FVM filter?

We can write discrete equations for \bar{u}^Δ with a resolved and unresolved part without applying the FVM filter f^h . For example, the LES equation $\overline{L(u)}^\Delta = 0$ can be rewritten as

$$L^h(\bar{u}^\Delta) = -(\partial_x \overline{r(u)}^\Delta - \partial_x^h r^h(\bar{u}^\Delta)). \quad (49)$$

The left-hand side is grid-compatible, and if the right-hand side is approximated by a grid-compatible closure model we do indeed get a discrete approximation of \bar{u}^Δ without applying the FVM filter. However, the residual in the right-hand side cannot be written as a discrete flux term (unless f^Δ is the FVM filter f^h), so this equation is not ∂_x^h -conservative. To obtain a discrete conservation law, the FVM filter must be applied.

3.6. Closure modeling implications

In LES, it is common to model the residual flux $\tau^\Delta(u)$ with artificial dissipation, as in eddy viscosity models such as the Smagorinsky model (10). For the FVM, it is common to incorporate artificial numerical dissipation directly into the numerical flux r^h [15], notably to prevent oscillations around sharp gradients and shocks [26]. When used to model subgrid scale effects, this approach is called *implicit* LES. Dissipation can also be explicitly added through a dissipative flux $m^h(\bar{u}^h) \approx \tau^h(u)$ on top of a DNS-like flux $r^h(\bar{u}^h)$ that has low numerical dissipation.

In our combined LES-FVM framework, the closure $m^{\Delta,h}$ should account for both LES and FVM effects. The advantage of including the numerical flux r^h in the definition of $\tau^{\Delta,h}$ is that the implicit dissipation produced by r^h is automatically accounted for in $\tau^{\Delta,h}$. This allows for better determining how much additional dissipation is required by the explicit closure $m^{\Delta,h}(\bar{u}^{\Delta,h}) \approx \tau^{\Delta,h}(u)$ [27]. If $m^{\Delta,h}$ is a Smagorinsky model, then the Smagorinsky constant can be tuned so that $m^{\Delta,h}(\bar{u}^{\Delta,h})$ and $\tau^{\Delta,h}(u)$ have similar dissipation profiles for a few reference snapshots u obtained from DNS. If r^h is a low-dissipation flux, such as the central difference based Burgers flux (15), the resulting Smagorinsky constant would be higher. If r^h already contains implicit numerical dissipation (as is common for upwinding fluxes), then the resulting Smagorinsky constant would be lower, since less additional explicit dissipation would be required.

In the classical LES-FVM approach “approximate first, then discretize”, the LES closure m^Δ is in principle made without knowledge of the discretization effects. As a remedy, information about the discretization is loosely reinjected into the definition of m^Δ by relating Δ to the grid size h .

Next, we turn to the Burgers equation to illustrate the consequences of the new LES-FVM framework. In Section 4.2, we investigate how the definition of the residual flux affects the LES-FVM equation. In Section 4.3, we investigate how the choice of “approximate first” vs “discretize first” leads to different closure models in practice.

4. LES-FVM For Burgers’ equation

We first consider the one-dimensional viscous Burgers’ equation defined by the flux (2). This equation can be seen as a simplified version of the Navier-Stokes equations, without the pressure term. In the inviscid limit, the solution forms shocks, which can cause oscillations with a coarse grid discretization with a low-dissipation numerical flux such as the one in Eq. (15) [26]. These issues also persist in the viscous case if the grid is too coarse. This is why discretization-aware closure models are needed.

To assess the correctness of the LES-FVM framework, we require reference solutions to Burgers’ equation. We use DNS to approximate the reference solution u and all derived quantities (such as $\bar{u}^{\Delta,h}$ and $\tau^{\Delta,h}(u)$) on a DNS grid of sufficiently small grid spacing $h_{\text{DNS}} \ll h$. In Appendix B, we show how the quantities in our LES-FVM framework can be computed from DNS data in a consistent manner (such that the filter-swap property still holds). This step involves discretizing the two infinitesimal filters f^Δ and f^h on the DNS grid. It also adds the constraint that the refinement factor h/h_{DNS} must be odd. Since the notation required to keep track of quantities related to both the fine h_{DNS} -grid and coarse h -grid becomes quite verbose, we will keep the infinitesimal notation for the reference solution u in the following, even though it is approximated on the h_{DNS} -grid in the code. Note also that h here refers to the coarse LES-FVM grid spacing, while in Appendix B, h refers to the fine DNS grid spacing.

4.1. Simulation setup

We use the following unitless parameters. The domain size is $\ell := 2\pi$. The viscosity is $\nu := 5 \times 10^{-4}$. The numerical flux is the central difference based flux in Eq. (15). The grid spacings are $h := \ell/N$ and $h_{\text{DNS}} := \ell/N_{\text{DNS}}$. We consider one DNS grid size $N_{\text{DNS}} := 13500$ and multiple LES grid sizes $N \in \{300, 900, 2700\}$. The compression factors are $N_{\text{DNS}}/N \in \{45, 15, 5\}$ respectively. These resolutions give an odd refinement factor for multiple LES grid sizes at a fixed DNS grid size. The filter-swap compatibility between the DNS and LES grids requires N_{DNS}/N to be an odd integer (see Appendix B). The LES filter kernel is the Gaussian (38). Unless otherwise stated, the LES filter width is $\Delta = 2h$.

The initial conditions for u are prescribed through the Fourier coefficients

$$\hat{a}(k) := \left(\frac{k}{k_0}\right)^4 \exp\left(-2\left(\frac{k}{k_0}\right)^2 + 2\pi i \epsilon_k\right), \quad (50)$$

where $\widehat{(\cdot)}$ is the Fourier transform, $k \in \mathbb{Z}$ is the wavenumber, i is the imaginary unit, $k_0 = 10$ is the peak wavenumber, and $\epsilon_k \sim \mathcal{U}(0, 1)$ is a uniformly sampled random number between 0 and 1 for $k \geq 0$ and $\epsilon_k = -\epsilon_{-k}$ otherwise. The velocity field is then normalized as

$$u := \sqrt{2E_0}a/\|a\|_{\Omega}, \quad (51)$$

where $E_0 := 2$ is the chosen initial total energy and $\|a\|_{\Omega}^2 := \int_{\Omega} |a|^2 dx$. This gives the initial energy spectrum profile $|\hat{u}(k)|^2 \propto k^4 e^{-2(k/k_0)^2}$ which is commonly used for decaying turbulence problems [1,28,29]. In total, we generate 1000 different initial conditions $u \in \mathcal{U}_{\text{initial}}$.

4.2. DNS-Aided LES-FVM

To evaluate the correctness of different residual fluxes, we employ the ‘‘DNS-aided LES(FVM)’’-framework of Bae and Lozano-Duran [6,30]. It consists of running a DNS alongside an LES-FVM, where the DNS solution is used to compute the closure term used in the LES-FVM equation. Ideally, by using the DNS solution to compute the right-hand side, one would expect to be able to recover the filtered DNS solution with the DNS-aided LES-FVM. We here solve the coupled DNS/LES-FVM equations

$$L(u) = 0, \quad L^h(v^{\Delta,h}) = -\partial_x^h \tau_{\text{model}}^{\Delta,h}(u), \quad (52)$$

where $\tau_{\text{model}}^{\Delta,h}(u)$ is a ‘‘closure’’ model computed from the DNS solution u . This term does not depend on the LES-FVM approximation $v^{\Delta,h}$. The LES-FVM approximation is initialized with $v^{\Delta,h} = \bar{u}^{\Delta,h}$.

We test four different expressions for $\tau_{\text{model}}^{\Delta,h}$:

$$\tau_{\text{no-model}}^{\Delta,h}(u) := 0, \quad (53)$$

$$\tau_{\text{classic}}^{\Delta,h}(u) := r(u)^{\Delta,h} - r(\bar{u}^{\Delta,h}), \quad (54)$$

$$\tau_{\text{classic+flux}}^{\Delta,h}(u) := r(u)^{\Delta,h} - r^h(\bar{u}^{\Delta,h}), \quad (55)$$

$$\tau_{\text{classic+flux+div}}^{\Delta,h}(u) := r(u)^{\Delta} - r^h(\bar{u}^{\Delta,h}). \quad (56)$$

From Eq. (35) and the decomposition (45), we know that

$$\tau_{\text{classic+flux+div}}^{\Delta,h} = \tau^{\Delta,h} = \tau_{\text{classic}}^{\Delta,h} + \tau_{\text{flux}}^{\Delta,h} + \tau_{\text{div}}^{\Delta,h} \quad (57)$$

is the correct expression. The two other non-zero models are obtained by neglecting the discretization errors: $\tau_{\text{classic}}^{\Delta,h}$ neglects both the divergence error $\tau_{\text{div}}^{\Delta,h}$ and the numerical flux error $\tau_{\text{flux}}^{\Delta,h}$, while $\tau_{\text{classic+flux}}^{\Delta,h} = \tau_{\text{classic}}^{\Delta,h} + \tau_{\text{flux}}^{\Delta,h}$ only neglects the divergence error $\tau_{\text{div}}^{\Delta,h}$.

Both u and $v^{\Delta,h}$ are advanced forward in time using the forward-Euler scheme

$$u_{k+1} := u_k - \Delta t_k \partial_x r(u_k) \quad (58)$$

$$v_{k+1}^{\Delta,h} := v_k^{\Delta,h} - \Delta t_k \partial_x \left(r^h(v_k^{\Delta,h}) + \tau_{\text{model}}^{\Delta,h}(u_k) \right) \quad (59)$$

where u_k and $v_k^{\Delta,h}$ denote the forward-Euler approximations to the DNS and LES-FVM solutions at time $t_k := \sum_{i=0}^{k-1} \Delta t_i$. The time step

$$\Delta t_k := C \times \min \left(\frac{h_{\text{DNS}}}{\max_{x \in \Omega} |u_k(x)|}, \frac{h_{\text{DNS}}^2}{v} \right) \quad (60)$$

is chosen based on the CFL condition for u_k with $C = 0.4$. Both equations use the same time step. While the time stepping scheme is of first order accuracy only, it has the advantage of only requiring one evaluation of the right-hand side per time step. This simplifies the injection procedure, where the residual flux from DNS is injected into the LES-FVM equation. It would be more complicated to inject the DNS solution into the LES-FVM equation for higher order schemes with multiple stages per time step.

In Fig. 4, we show the DNS solution for one initial condition at initial and final time ($t = 0.1$). The solution is slightly damped due to dissipation, and some shock-like structures are forming. These sharp gradients cannot be properly resolved on the coarse h -grid and cause oscillations with the central difference ∂_x^h , which is why a discretization-informed closure model is needed.

In Fig. 5, we show the average fractions

$$\frac{1}{1000} \sum_{u \in \mathcal{U}_{\text{final}}} \frac{|\tau_{\text{label}}^{\Delta,h}(u)|}{|\tau_{\text{classic}}^{\Delta,h}(u)| + |\tau_{\text{flux}}^{\Delta,h}(u)| + |\tau_{\text{div}}^{\Delta,h}(u)|} \quad (61)$$

for label $\in \{\text{classic, flux, div}\}$, where $\mathcal{U}_{\text{final}}$ contains the DNS solutions at the final time. For $\Delta = 0$, the classic flux only accounts for about 28% of the total residual flux. For $\Delta = 4h$, the classic flux accounts for about 73–79% of the total. For $\Delta = 32h$, less than 1% of the residual is due to the discretization errors $\tau_{\text{flux}}^{\Delta,h}$ and $\tau_{\text{div}}^{\Delta,h}$. For explicit LES, at $\Delta \geq 32h$, we can therefore safely neglect the discretization errors in the residual flux. For $\Delta \leq 4h$, the discretization errors are significant and should not be neglected. This corresponds with the observation of Chow and Moin that $\Delta \geq 4h$ is necessary for the second order discretization errors to no longer dominate the residual flux [8]. For $4h < \Delta < 32h$, the discretization errors may or may not be neglected depending on what accuracy is required. For an actual closure model, which is going to have a significant modeling error anyway, the 21% contribution of the

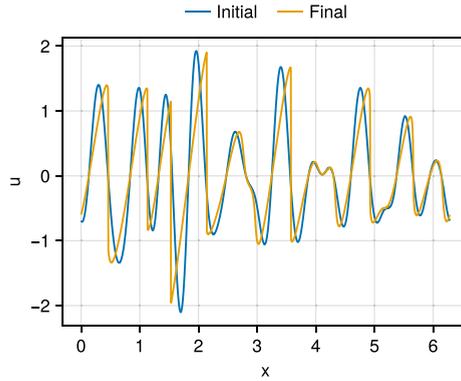


Fig. 4. Initial and final solution to the Burgers equation.

discretization to the residual flux at $(\Delta, N) = (4 h, 300)$ may not be important. But if the residual flux is to be used as a target for tuning the closure model coefficients, including the 6 % contribution of the discretization at $(\Delta, N) = (8 h, 300)$ could still be useful.

Fig. 5 only shows the importance of the discretization errors in the a-priori setting. In Fig. 6, we show the average a-posteriori relative errors

$$e_{\text{model}} := \frac{1}{1000} \sum_{u_0 \in \mathcal{U}^{\text{initial}}} \frac{\|v_{\text{model}}^{\Delta, h} - \bar{u}^{\Delta, h}\|_{\Omega}}{\|\bar{u}^{\Delta, h}\|_{\Omega}} \tag{62}$$

at the final time for different values of N and Δ . For all N and Δ , $e_{\text{classic} + \text{flux} + \text{div}}$ is at machine precision. For $(N, \Delta) = (300, 0 h)$, we have $e_{\text{no-model}} \approx 60\%$, $e_{\text{classic}} \approx 16\%$, and $e_{\text{classic} + \text{flux}} \approx 8\%$. For $(N, \Delta) = (300, 4 h)$, we have $e_{\text{no-model}} \approx 65\%$, $e_{\text{classic}} \approx 7\%$, and $e_{\text{classic} + \text{flux}} \approx 2\%$. For $(N, \Delta) = (300, 32 h)$, we have $e_{\text{no-model}} \approx 35\%$, $e_{\text{classic}} \approx 0.1\%$, and $e_{\text{classic} + \text{flux}} \approx 0.03\%$. This confirms that the discretization errors are significant for $\Delta \leq 4h$, and insignificant for $\Delta \geq 32h$.

Further insight in the four models is obtained through the energy spectrum. We define the energy spectrum of a field u as $|\hat{u}(k)|^2/2$, where $\hat{u}(k)$ is the Fourier transform of u at wavenumber k . Fig. 7 shows the energy spectra at the final time, averaged over the 1000 solutions. Individual DNS solutions have noisy spectra that fluctuate around the theoretical slope of k^{-2} in the inertial range. The averaged DNS spectrum is smooth, and adheres to the theoretical slope for the inertial range. The DNS grid spacing is small enough to resolve both the dissipation and inertial ranges.

The filtered DNS spectrum stays on top of the DNS spectrum for the low wavenumbers, and becomes damped for the highest wavenumbers. This is because the transfer function of the filter $f^{\Delta, h}$ is close to 1 for low wavenumbers, but decays for larger wavenumbers (see Fig. 3). The filtered DNS spectra are shown beyond the cut-off wavenumbers $N/2$ of the coarse h -grids. These spectra are obtained by evaluating $\bar{u}^{\Delta, h}(x)$ at all the DNS grid points x , and then applying the Fourier transform on the DNS grid. Since $f^{\Delta, h}$ is not a spectral cut-off filter, the spectrum of $\bar{u}^{\Delta, h}(x_i^h)$ restricted to the coarse h -grid points x_i^h , computed with a coarse h -grid discrete Fourier transform, is slightly different from the spectrum of $\bar{u}^{\Delta, h}(x)$ computed on the DNS grid. In the more detailed zoom-in box, we therefore show the spectrum of $\bar{u}^{\Delta, h}(x_i^h)$, which is not defined beyond the cut-off wavenumber $N/2$ of the h -grid.

The LES-FVM spectra are all evaluated on the h -grid, and they are therefore not defined beyond $N/2$. For all three grid sizes, the no-closure solution has too much energy in the highest resolved wavenumbers. This is common in LES, and part of the motivation for using a dissipative closure model. The classic flux $\tau_{\text{classic}}^{\Delta, h}$ is not dissipative enough. For the first two resolutions, it even shows signs of instability in the highest wavenumbers. The improved flux $\tau_{\text{classic} + \text{flux}}^{\Delta, h}$ gives a spectrum closer to the filtered DNS than $\tau_{\text{classic}}^{\Delta, h}$, but at the highest resolved wavenumbers, it is generally *too dissipative*. It also shows a sign of instability at the highest wavenumbers. This is likely because the DNS-aided closure term $\tau_{\text{classic} + \text{flux}}^{\Delta, h}(u)$ does not depend on the LES-FVM solution $v_{\text{classic} + \text{flux}}^{\Delta, h}$ at all. As $v_{\text{classic} + \text{flux}}^{\Delta, h}$ starts to decorrelate from $\bar{u}^{\Delta, h}$, the DNS-aided closure term becomes less accurate over time and even detrimental at the highest wavenumbers. The correct flux $\tau_{\text{classic} + \text{flux} + \text{div}}^{\Delta, h}$ gives a spectrum that perfectly overlaps with the h -grid variant of the filtered DNS spectrum (in the zoom-in box).

Conclusion. Using the correct expression for the residual LES-FVM flux as a closure model gives *perfect* results, unlike the classical residual LES flux expression. The exact residual flux $\tau_{\text{classic} + \text{flux} + \text{div}}^{\Delta, h}(u) = \tau^{\Delta, h}(u)$ can therefore be seen as the *best case scenario* for an LES-FVM closure model $m^{\Delta, h}(\bar{u}^{\Delta, h})$. However, in general $\tau^{\Delta, h}(u)$ cannot be computed from $\bar{u}^{\Delta, h}$ alone, since information is lost in the filtering process.

Next, we turn to an actual LES-FVM closure model that does not require a parallel DNS simulation for evaluation. The question is whether the discretization-informed expression for the residual flux can actually be used to make better closure models.

4.3. Smagorinsky closure

Define the infinitesimal and discrete Smagorinsky fluxes as

$$s(u) := |\partial_x u| \partial_x u, \quad s^h(u) := \left| \partial_x^h u \right| \partial_x^h u. \tag{63}$$

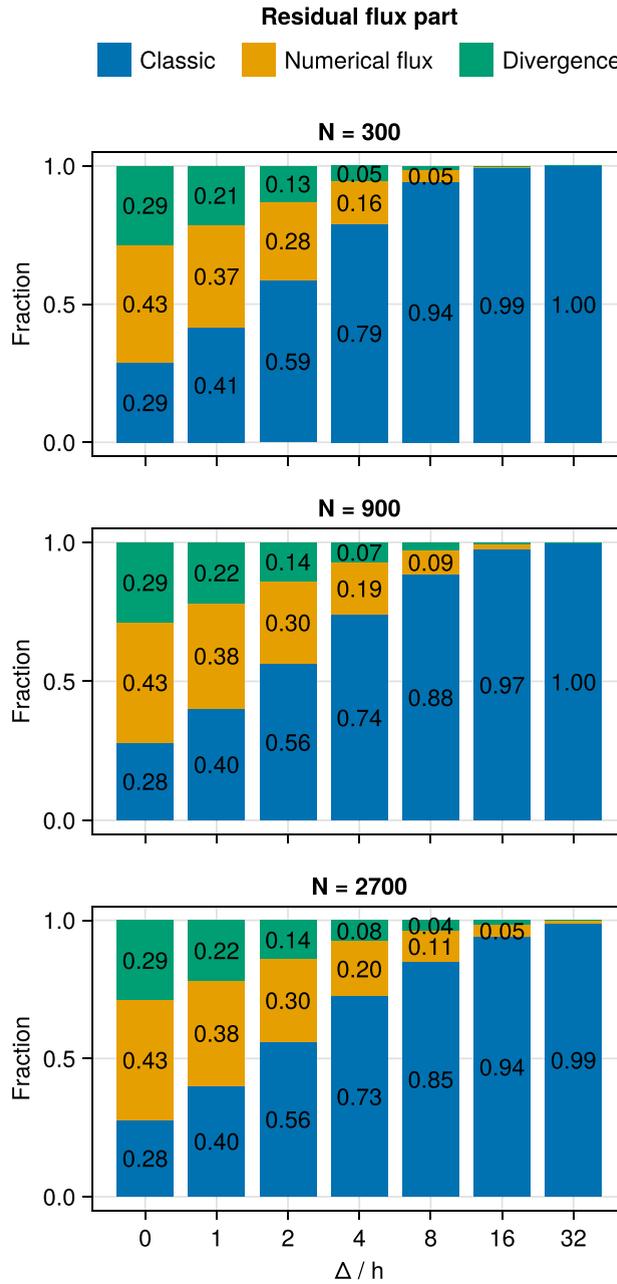


Fig. 5. Relative contribution of the different flux parts in the decomposition (45).

For the double-filter $f^{\Delta,h}$ of width $\sqrt{\Delta^2 + h^2}$, the classical LES Smagorinsky model is defined at the infinitesimal level as

$$m_{\text{classic}}^{\Delta,h}(u; \theta) := -\theta^2(\Delta^2 + h^2)s(u), \tag{64}$$

where θ is a model parameter. This classical LES model is then discretized as

$$m_{\text{discrete}}^{\Delta,h}(u; \theta) := -\theta^2(\Delta^2 + h^2)s^h(u). \tag{65}$$

The coefficient θ can be estimated in various ways. For the infinitesimal basic Smagorinsky model (before discretizing the flux), a value can be derived analytically [31]. In the dynamic Smagorinsky model, the coefficient is estimated from the LES state itself using a hypothesis of scale-similarity combined with a test-filter [21]. We employ the basic Smagorinsky framework, but with a data-driven estimate of the coefficient using a least-squares fit to the residual flux.

In the classical LES framework, the infinitesimal model $m_{\text{classic}}^{\Delta,h}(\bar{u}^{\Delta,h}; \theta)$ is used to approximate the residual LES flux $\tau_{\text{classic}}^{\Delta,h}(u) := \overline{r(u)^{\Delta,h}} - r(\bar{u}^{\Delta,h})$. The discretization is then only accounted for in the filter definition. For a set of reference solution snapshots \mathcal{U}

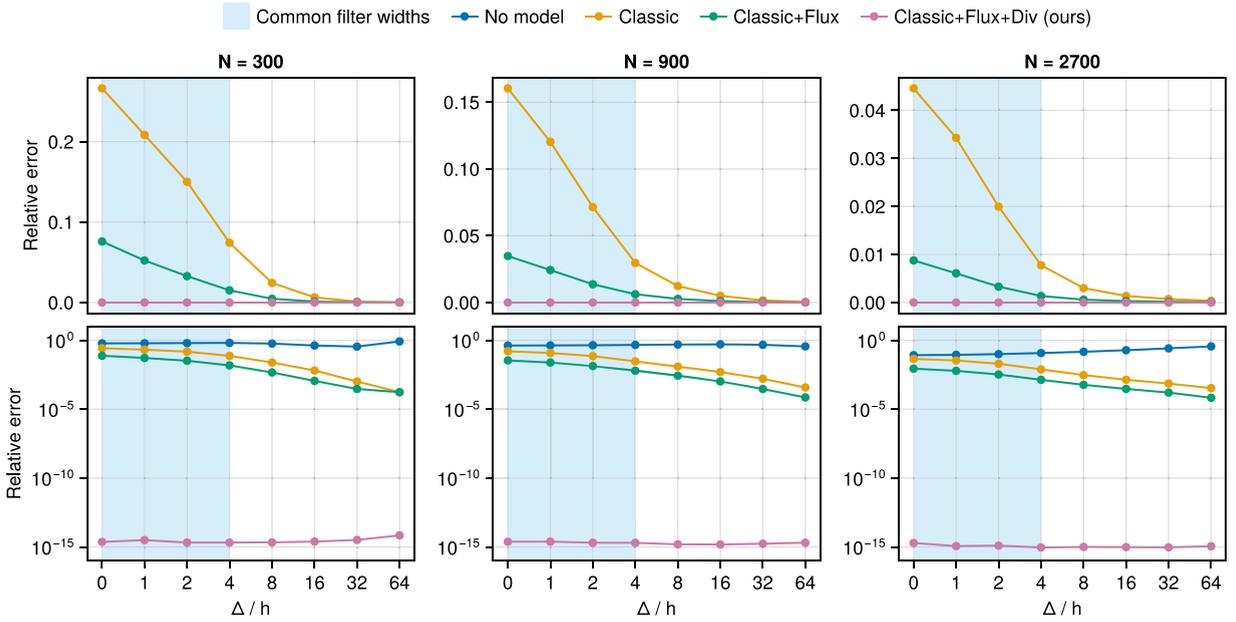


Fig. 6. Relative errors at final time for DNS-aided LES-FVM of Burgers' equation. The shaded area indicates common ratios Δ/h in literature. Top: Linear scale (the no-model error is above the plotted region). Bottom: Logarithmic scale.

(obtained from DNS), we therefore estimate the coefficient for the classical Smagorinsky model as

$$\min_{\theta} \sum_{u \in \mathcal{U}'} \left| -\theta^2 (\Delta^2 + h^2) s(\bar{u}^{\Delta,h}) - \tau_{\text{classic}}^{\Delta,h}(u) \right|^2, \tag{66}$$

where the infinitesimal operators are approximated on the DNS grid. Similarly, in the new LES-FVM framework, a discretization-informed Smagorinsky model can be obtained by choosing θ such that $m_{\text{discrete}}^{\Delta,h}(\bar{u}^{\Delta,h}; \theta)$ matches the discretization-informed residual flux $\tau^{\Delta,h}(u) := \overline{r(u)}^{\Delta} - r^h(\bar{u}^{\Delta,h})$ as

$$\min_{\theta} \sum_{u \in \mathcal{U}'} \left| -\theta^2 (\Delta^2 + h^2) s^h(\bar{u}^{\Delta,h}) - \tau^{\Delta,h}(u) \right|^2. \tag{67}$$

We thus get the two different least-squares estimates for θ :

$$-\theta_{\text{classic}}^2 (\Delta^2 + h^2) = \frac{\sum_{u \in \mathcal{U}'} \langle s(\bar{u}^{\Delta,h}), \tau_{\text{classic}}^{\Delta,h}(u) \rangle}{\sum_{u \in \mathcal{U}'} \langle s(\bar{u}^{\Delta,h}), s(\bar{u}^{\Delta,h}) \rangle}, \tag{68}$$

$$-\theta_{\text{informed}}^2 (\Delta^2 + h^2) = \frac{\sum_{u \in \mathcal{U}'} \langle s^h(\bar{u}^{\Delta,h}), \tau^{\Delta,h}(u) \rangle}{\sum_{u \in \mathcal{U}'} \langle s^h(\bar{u}^{\Delta,h}), s^h(\bar{u}^{\Delta,h}) \rangle}, \tag{69}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product on Ω . Solving for θ is then straightforward. These expressions are similar to the ones in the dynamic Smagorinsky model (where the coefficient is also tuned using a least-squares estimate), but here we use DNS data u instead of LES data to compute the target residual flux.

What is the difference between θ_{classic} and θ_{informed} ? In the classical LES setting, θ_{classic} is obtained *before* discretization (information about the discretization is only injected through the definition of the filter width). In our framework, θ_{informed} is obtained *after* discretization. It should therefore be slightly higher than θ_{classic} to account for the additional dissipation required by the discretization. Our hypothesis is that the classical discretized model $m_{\text{discrete}}^{\Delta,h}(\cdot; \theta_{\text{classic}})$ should perform worse than the discretization-informed model $m_{\text{discrete}}^{\Delta,h}(\cdot; \theta_{\text{informed}})$, since the choice of θ_{classic} assumes an infinitesimal setting, while θ_{informed} is obtained in the discrete setting, using the correct discrete residual flux as a target.

In Fig. 8, we show the Smagorinsky coefficients estimated from DNS data for Gaussian f^{Δ} and $\Delta \in \{0h, 2h\}$. Indeed, θ_{discrete} is larger than θ_{classic} for all grid sizes. The two coefficients are more different for $\Delta = 0h$ than for $\Delta = 2h$, likely because $\tau^{\Delta,h}$ converges to $\tau_{\text{classic}}^{\Delta,h}$ when the ratio Δ/h increases (see Fig. 5). For even larger Δ , for example $\Delta = 32h$, the two coefficients would likely be very similar.

Since all three grid sizes lay within the inertial range of the energy spectrum, we could also choose to fit one single Smagorinsky coefficient for all grid sizes. Currently, we fit separate coefficients for each grid size N and filter ratio Δ/h , resulting in larger coefficients for the smaller grid sizes and filter ratios. This is likely because the sharp gradients are less resolved, and therefore more dissipation is needed.

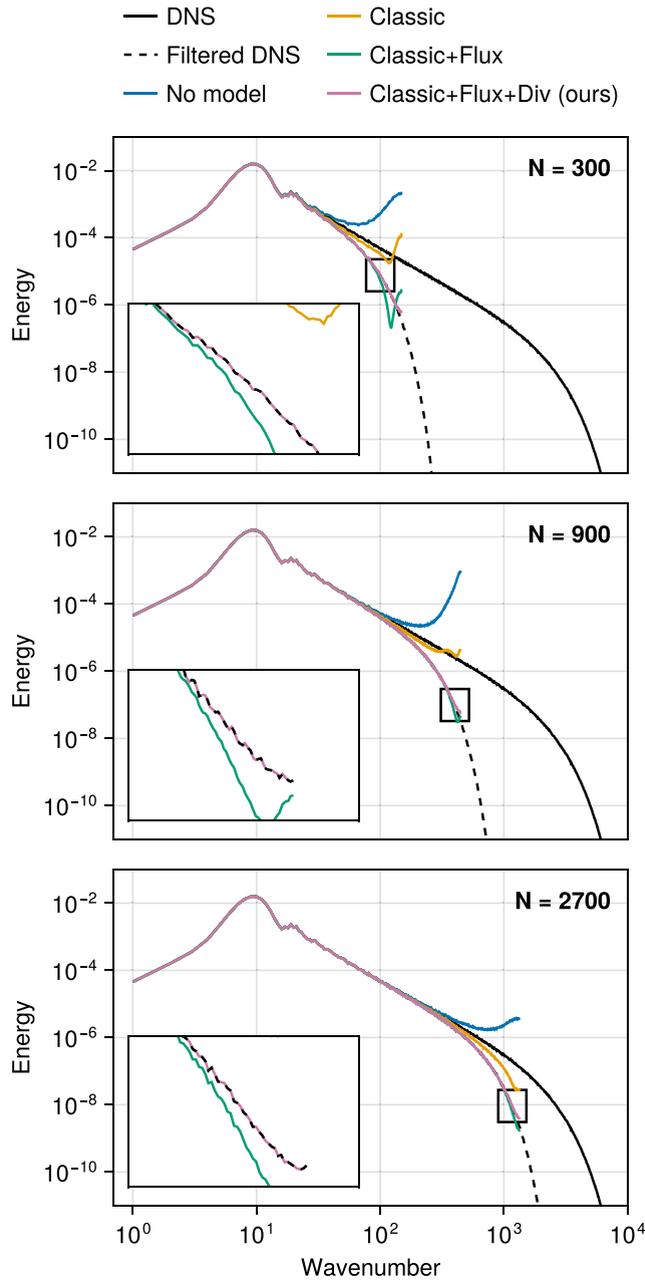


Fig. 7. Energy spectra for the DNS-aided LES-FVM of Burgers equation. Our discretization-informed expression leads to an exact correspondence with the filtered DNS, while the classical expression is not dissipative enough.

In Fig. 9, we show the relative errors at the final time between the Smagorinsky LES-FVM approximation and the exact LES-FVM solution. Our discretization-informed Smagorinsky model has lower errors than the classical Smagorinsky model. The difference is significant for $\Delta = 0h$, but smaller for $\Delta = 2h$. For even larger filter ratios, the difference would likely vanish.

The difference is further visible in the energy spectra shown in Fig. 10. We see that the classical Smagorinsky model is not dissipative enough with a larger spectrum than the filtered DNS. The discretization-informed Smagorinsky model has a spectrum closer to the filtered DNS. For $\Delta = 2h$, both models have a peak at the highest wavenumbers. This is because the central difference scheme used in r^h gives rise to oscillations around the sharp gradients, and the dissipative Smagorinsky model is not able to fully remove them without causing over-dissipation across all wavenumbers. The Smagorinsky coefficient obtained from the least-squares fit is a compromise between removing these oscillations and preserving the rest of the spectrum.

We stress that the gain in accuracy obtained by tuning the Smagorinsky model with the correct residual flux instead of the classic one, is strongly limited by the eddy viscosity hypothesis. We expect that with more expressive closure models, such as machine-

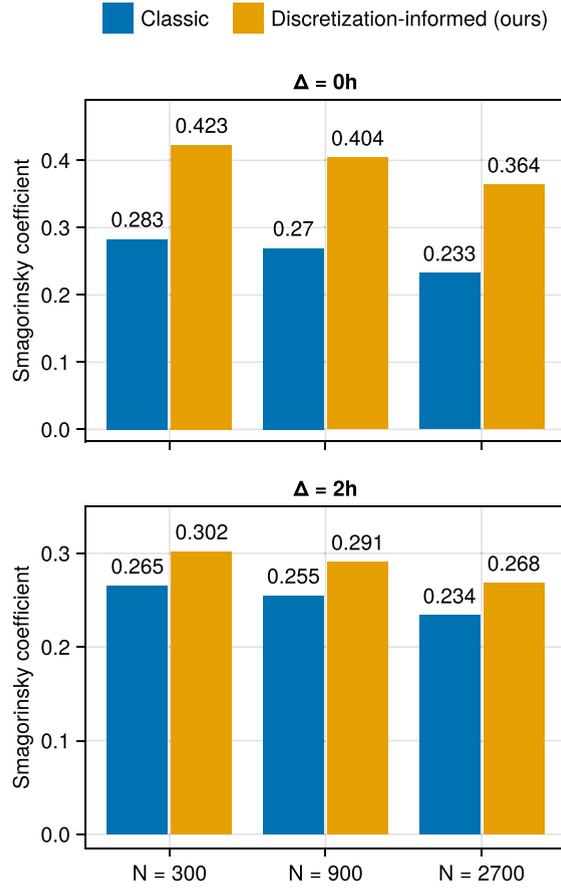


Fig. 8. Fitted Smagorinsky coefficients for Burgers' equation.

learning-based ones, much lower errors can be obtained, and the benefits of our new residual flux expression will be much more prominent.

In conclusion, we have shown that for common filter ratios, the discretization-informed residual flux for the 1D Burgers equation both leads to zero error in a DNS-aided LES and improves the performance of an actual closure model when used as a target for fitting the model parameters. Next, we turn to the 3D incompressible Navier-Stokes equations. As we show, they have additional complexities that must be overcome to achieve a discretization-informed expression for the residual.

5. LES-FVM For the 3D incompressible navier-Stokes equations

The incompressible Navier-Stokes equations in index-form are given by

$$\partial_j u_j = 0, \quad \partial_t u_i + \partial_j (\sigma_{ij}(u) + p \delta_{ij}) = 0, \quad (70)$$

where

$$\sigma_{ij}(u) := u_i u_j - \nu (\partial_j u_i + \partial_i u_j) \quad (71)$$

is the convective-diffusive stress tensor, δ_{ij} is the identity tensor (Kronecker delta-symbol), $(i, j) \in \{1, 2, 3\}^2$ are indices, $\partial_t := \partial/\partial t$ and $\partial_i := \partial/\partial x_i$ are partial derivatives, $x := (x_1, x_2, x_3)$ is the position, t is the time, $u_i(x, t)$ is the velocity in direction i , $p(x, t)$ is the pressure, and $\nu > 0$ is the kinematic viscosity. Repeated indices imply summation (Einstein notation). Since our derivations involve mainly spatial derivatives, we will write $u_i(x)$ instead of $u_i(x, t)$ to ease the notation. For simplicity, we assume the equations are defined in a periodic box $\Omega = [0, \ell]^3$ of side length $\ell > 0$.

The space of periodic scalar-valued fields on Ω is denoted U . Although the entries of the tensor $\sigma(u) \in U^{3 \times 3}$ are scalar fields $\sigma_{ij}(u) \in U$, we still use the word “tensor” to describe both $\sigma(u)$ and $\sigma_{ij}(u)$ interchangeably (and similarly for other vector and tensor fields).

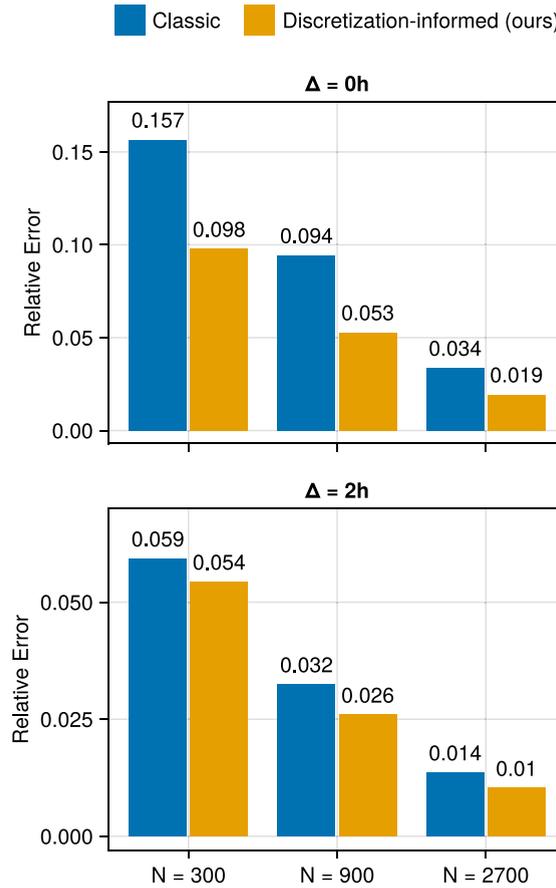


Fig. 9. Relative errors at final time for LES-FVM of Burgers' equation with the Smagorinsky model.

5.1. Pressure-free navier-Stokes equations

The system (70) does not have the form of a conservation law for the state $(u, p) \in U^4$. To repeat the procedure from the previous sections (1D), we first need to rewrite the Navier-Stokes equations as a self-contained conservation-law. To achieve this, we will eliminate the pressure p from the momentum equations.

The pressure p is a Lagrange multiplier that enforces the continuity equation for u . A related viewpoint is that $p\delta_{ij}$ is a correction that makes the non-divergence-preserving stress tensor $\sigma_{ij}(u)$ divergence-preserving. We say that a stress tensor $\sigma \in U^{3 \times 3}$ is divergence-preserving if $\partial_i \sigma_{ij} = 0$, i.e. the force $\partial_j \sigma_{ij}$ is divergence-free.

In Appendix D, we introduce the two pressure projection operators

$$\pi_{ij} := \delta_{ij} - \partial_i (\partial_k \partial_k)^{\dagger} \partial_j \tag{72}$$

and

$$\pi_{ij\alpha\beta} := \delta_{i\alpha} \delta_{j\beta} - \delta_{ij} (\partial_k \partial_k)^{\dagger} \partial_{\alpha} \partial_{\beta}, \tag{73}$$

where $(\partial_k \partial_k)^{\dagger}$ is the inverse Laplacian operator subject to the constraint of an average pressure of zero. The vector-projector π_{ij} can be used to make vector fields divergence-free (since $\partial_i \pi_{ij} = 0$), while the tensor-projector $\pi_{ij\alpha\beta}$ can be used to make tensor fields divergence-preserving (since $\partial_i \partial_j \pi_{ij\alpha\beta} = 0$). The proofs are given in Theorem 6 and Theorem 7.

Since u is divergence-free, we can rewrite the momentum equation in a pressure-free way using either of the two projectors as $\partial_t u_i + \pi_{ij} \partial_k \sigma_{jk}(u) = 0$ or $\partial_t u_i + \partial_j \pi_{ij\alpha\beta} \sigma_{\alpha\beta}(u) = 0$. We will use the latter form, since it has the form of a conservation law (divergence of a tensor). The pressure-free momentum equations can thus be written as

$$L_i(u) := \partial_t u_i + \partial_j r_{ij}(u) = 0, \tag{74}$$

where

$$r_{ij}(u) := \pi_{ij\alpha\beta} \sigma_{\alpha\beta}(u) = \sigma_{ij}(u) + p\delta_{ij} \tag{75}$$

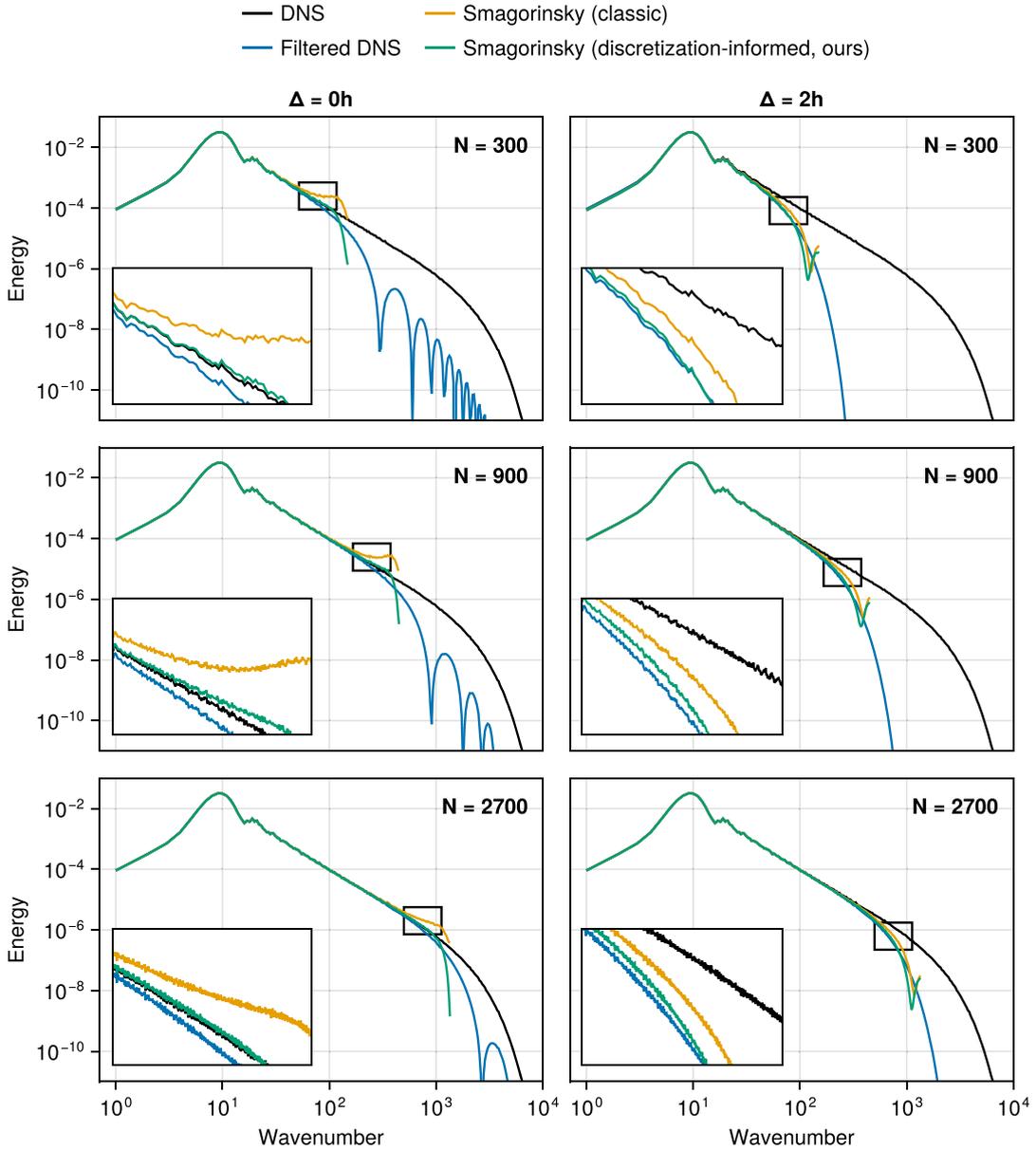


Fig. 10. Energy spectra for the Burgers equation solved with the Smagorinsky model.

is the projected stress tensor and $L_i := \partial_t + \partial_j r_{ij}$ is the pressure-free momentum equation operator. Given $\sigma(u)$, the projector π computes the unique pressure p (up to a constant) such that $\sigma(u) + p\delta$ is divergence-preserving. The pressure projection only modifies the diagonal, so $r_{ij} = \sigma_{ij}$ for $i \neq j$.

In Eq. (70), there is a spatial constraint of divergence-freeness, which was not present in Section 2. In the projected form (74), this constraint is hidden inside $r_{ij}(u)$. As a result, the 3D stress tensor $r_{ij}(u)$ is non-local in u , and requires solving a Poisson equation, whereas the 1D Burgers flux $r(u)$ was local. As long as the initial velocity field is divergence-free, the continuity equation can be ignored since Eq. (74) evolves the velocity field in a divergence-preserving way.

Since we incorporated the divergence-free constraint, the 3D conservation law (74) has the same form as the 1D conservation law (1). We can therefore repeat the procedure from Section 3 to obtain LES-FVM equations in conservative form. We use the same notation as in Sections 2, 3, and 4 to highlight the similarities and differences. One difference is the presence of direction indices i and j . The scalar flux $r(u)$ from Section 2 is now a 3×3 stress tensor $r_{ij}(u)$.

5.2. Classical LES

Consider a convolutional homogeneous spatial filter $f^\Delta : u \mapsto \bar{u}^\Delta$ defined for all scalar fields $u \in U$ as

$$\bar{u}^\Delta(x) := \int_{\mathbb{R}^3} G^\Delta(x-y)u(y) dy \tag{76}$$

for some kernel G^Δ . We integrate over \mathbb{R}^3 instead of Ω to allow for periodic extension. As in 1D, this filter commutes with differentiation:

$$f^\Delta \partial_i = \partial_i f^\Delta. \tag{77}$$

The filtered Navier-Stokes equations therefore take the conservative form

$$\partial_j \bar{u}_j^\Delta = 0, \quad \partial_i \bar{u}_i^\Delta + \partial_j \left(\sigma_{ij}(\bar{u}^\Delta) + \xi_{ij}^\Delta(u) + \bar{p}^\Delta \delta_{ij} \right) = 0, \tag{78}$$

where \bar{u}_i^Δ and \bar{p}^Δ are filtered fields and

$$\xi_{ij}^\Delta(u) := \overline{u_i u_j}^\Delta - \bar{u}_i^\Delta \bar{u}_j^\Delta \tag{79}$$

is the classical RST. We reserve the symbol τ^Δ for the RST $\tau^\Delta(u) := \pi \xi^\Delta(u) = \overline{r(u)^\Delta} - r(\bar{u}^\Delta)$, which contains the pressure projector π . Since filtering commutes with differentiation, we also have $f^\Delta \pi = \pi f^\Delta$.

For classical structural LES models, the unprojected tensor $\xi_{ij}^\Delta(u)$ is replaced by a closure model $m_{ij}^\Delta(\bar{u}^\Delta)$ that only depends on \bar{u}^Δ . Since ξ_{ij}^Δ is a symmetric tensor ($\xi_{ij}^\Delta = \xi_{ji}^\Delta$), the closure model is designed to be symmetric as well ($m_{ij}^\Delta = m_{ji}^\Delta$). If the filter f^Δ is local in space, then ξ^Δ is also local, and m^Δ can be chosen to be a local closure. To solve the LES equations, the closed equations are discretized. Since m^Δ is symmetric, its discretized variant is also symmetric.

5.3. Classical FVM

Using the staggered spatial discretization scheme of Harlow and Welch [32], we define the DNS equations as

$$\partial_j^h u_j^h = 0, \quad \partial_i u_i^h + \partial_j^h \left(\sigma_{ij}^h(u^h) + p^h \delta_{ij} \right) = 0. \tag{80}$$

Here, $u^h \in U^3$ and $p^h \in U$ are the DNS velocity and pressure fields,

$$\sigma_{ij}^h(u) := \left(\eta_j^h u_i \right) \left(\eta_i^h u_j \right) - v \left(\partial_i^h u_i + \partial_i^h u_j \right) \tag{81}$$

is a grid-compatible numerical stress tensor analogous to the 1D numerical Burgers flux r^h from Eq. (15), $\partial_i^h : U \rightarrow U$ and $\eta_i^h : U \rightarrow U$ are finite difference and interpolation operators defined for all $u \in U$ as

$$\partial_i^h u(x) := \frac{1}{h} \left[u \left(x + \frac{h}{2} e_i \right) - u \left(x - \frac{h}{2} e_i \right) \right], \tag{82}$$

$$\eta_i^h u(x) := \frac{1}{2} \left[u \left(x - \frac{h}{2} e_i \right) + u \left(x + \frac{h}{2} e_i \right) \right], \tag{83}$$

where h is a uniform grid spacing and $(e_i)_{i=1}^3$ are the unit vectors. These operators are second-order accurate in h , and, as a result, $u^h(x, t)$ is a second-order accurate approximation of $u(x, t)$ for all x and t if $u^h(x, 0) = u(x, 0)$.

We use the projected form of the DNS Eq. (80):

$$L_i^h(u^h) := \partial_i u_i^h + \partial_j^h r_{ij}^h(u^h) = 0, \tag{84}$$

where L_i^h is the pressure-free finite volume momentum equation operator,

$$r_{ij}^h := \pi_{ij\alpha\beta}^h \sigma_{\alpha\beta}^h \tag{85}$$

is the projected discrete stress tensor, and

$$\pi_{ij\alpha\beta}^h := \delta_{i\alpha} \delta_{j\beta} - \delta_{ij} \left(\partial_k^h \partial_k^h \right)^\dagger \partial_\alpha^h \partial_\beta^h \tag{86}$$

is a discrete version of $\pi_{ij\alpha\beta}$ that makes stress tensors discretely divergence-preserving (i.e. $\partial_i^h \partial_j^h \pi_{ij\alpha\beta}^h = 0$). Given the stress $\sigma_{ij}^h(u^h)$, the projector $\pi_{ij\alpha\beta}^h$ computes the isotropic pressure correction $p^h \delta_{ij}$ that makes $r_{ij}^h(u^h)$ divergence-preserving, so we get the identity $r_{ij}^h(u^h) = \sigma_{ij}^h(u^h) + p^h \delta_{ij}$.

We divide the domain $\Omega = [0, \ell]^3$ into $N := \ell/h$ reference volumes in each dimension (N^3 volumes in total). When restricting the FVM solution, we use a staggered representation as depicted in Fig. 11. The pressure p^h is restricted to the volume centers (pressure points). The velocity components u_i^h are restricted to the centers of the volume faces orthogonal to e_i (velocity points). The positioning of the tensor components σ_{ij}^h follows naturally. They are in the pressure points if $i = j$, and in the centers of the volume edges otherwise. The continuity equation is evaluated in the pressure points, and the momentum equations in the velocity points.

Note that while we still use the continuous notation $u^h(x, t)$, the degrees of freedom depicted in Fig. 11 contain all the information needed to evaluate the DNS equations in the required points. This is because σ_{ij}^h is chosen such that the restricted DNS equations are closed in the discrete sense.

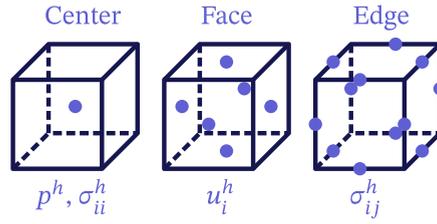


Fig. 11. Staggered positions in a reference volume of a scalar p^h , vector u^h , and tensor σ^h . The diagonal tensor components are in the volume center, while the off-diagonal tensor components are on the volume edges.

We use the divergence-form for the convective term, which is energy-conservative if $\partial_j^h u_j^h = 0$ [33]. The continuity equation $\partial_j^h u_j^h = 0$ is therefore enforced strictly by using semi-explicit time discretization schemes (applying a pressure projection *after* each momentum time step) [34,35].

5.4. Grid filters and the filter-swap property in 3D

To obtain LES-FVM equations in conservative form, we need grid filters that satisfy a filter-swap property. As in the 1D case, the finite difference operator ∂_i^h is associated to a one-dimensional top-hat filter $g_i^h : U \rightarrow U$ that is only applied in the direction e_i . For all $u \in U$, we define it as

$$g_i^h u(x) := \frac{1}{h} \int_{-h/2}^{h/2} u(x + \alpha e_i) d\alpha. \quad (87)$$

We can use the fundamental theorem of calculus to show the relation between ∂_i^h and g_i^h (with no summation over i):

$$\partial_i^h = g_i^h \partial_i, \quad (88)$$

meaning that the finite difference ∂_i^h can be induced by filtering the exact derivative ∂_i .

The Navier-Stokes momentum and continuity equations include derivatives in each of the cardinal directions x_1 , x_2 , and x_3 . For example, the i -th momentum equation includes the term $\partial_j r_{ij} = \partial_1 r_{i1} + \partial_2 r_{i2} + \partial_3 r_{i3}$. If we filter the i -th momentum equation with the 1D grid filter g_i^h in the direction i , the equation for $g_i^h u_i$ would include the term $g_i^h \partial_j r_{ij}$, and we could only use the filter-swap property on one of the three terms, where $j = i$. A similar remark was made by Lund, who argued that ideally, we would like to filter each of the three derivatives ∂_j with their associated filters g_j^h separately, but such equations cannot be obtained by applying one single filter to the Navier-Stokes momentum equations, since the same filter has to be applied to all of the terms [19]. We therefore resort to multi-dimensional grid filters.

From the 1D filters g_i^h , we define the multi-dimensional volume-averaging filter

$$f^h := g_1^h g_2^h g_3^h \quad (89)$$

and surface-averaging filters

$$f^{h,1} := g_2^h g_3^h, \quad f^{h,2} := g_1^h g_3^h, \quad f^{h,3} := g_1^h g_2^h. \quad (90)$$

For all $u \in U$, we employ the short-hand notation

$$\bar{u}^h := f^h u, \quad \bar{u}^{h,i} := f^{h,i} u. \quad (91)$$

A similar notation was used by Schumann [18].

By using the 1D property (88) for g_i^h , we obtain the following filter-swap commutation property for the multi-dimensional grid filters (with no sum over i):

$$f^h \partial_i = \partial_i^h f^{h,i}. \quad (92)$$

This is a discrete equivalent of the infinitesimal property $f^\Delta \partial_i = \partial_i f^\Delta$ which allows for switching between infinitesimal and discrete derivatives. The important observation is that the filter definition changes with the derivative definition. The 1D case in Eq. (19) can be seen as a special case of Eq. (92), where the 1D volume is $[x \pm h/2]$ and the 1D surface is a single point in $\{x \pm h/2\}$.

5.5. LES-FVM

The LES equation is $\overline{L(u)}^\Delta = 0$. Applying the FVM filter f^h gives the LES-FVM equation

$$\overline{L(u)}^{\Delta,h} = 0. \quad (93)$$

Using the filter-swap commutation property (92), we obtain the ∂_j^h -conservative form

$$\partial_i \bar{u}_i^{\Delta,h} + \partial_j^h \overline{r_{ij}^{\Delta,h,j}} = 0. \quad (94)$$

We can already note a couple of interesting observations. The stress tensor $\overline{r_{ij}(u)}^{\Delta,h,j}$ (no sum over j) is non-symmetric (because it is averaged over the surface orthogonal to j , but not i) and non-local in u (due to the projection present inside r). Furthermore, the LES-FVM solution $\bar{u}^{\Delta,h}$ is not discretely divergence-free, since $\partial_j^h \bar{u}_j^{\Delta,h} \neq \overline{\partial_j u_j}^{\Delta,h} = 0$ (see Theorem 2).³ Since the divergence-free constraint is important for stability in the energy-conserving convective term, we consider the divergence-free part of the volume-averaged field $\bar{u}^{\Delta,h}$. This is precisely the projected field

$$\bar{u}_i^{\Delta,h,\pi} := \pi_{ij}^h \bar{u}_j^{\Delta,h}. \quad (95)$$

Applying the projector π_{ij}^h to Eq. (94) and using Theorem 8 to swap projection and divergence gives

$$\partial_i \bar{u}_i^{\Delta,h,\pi} + \partial_j^h \pi_{ij\alpha\beta}^h \overline{r_{\alpha\beta}(u)}^{\Delta,h,\beta} = 0, \quad (96)$$

which is a discrete conservation law for a divergence-free velocity field.

Adding the discrete flux term $\partial_j^h r_{ij}^h(\bar{u}^{\Delta,h,\pi})$ to both sides gives the projected LES-FVM equation in the form

$$L_i^h(\bar{u}^{\Delta,h,\pi}) = -\partial_j^h \tau_{ij}^{\Delta,h,\pi}(u), \quad (97)$$

where the projected RST $\tau_{ij}^{\Delta,h,\pi} := \pi_{ij\alpha\beta}^h \xi_{\alpha\beta}^{\Delta,h,\pi}$ is defined by the RST (with no sum over j)

$$\xi_{ij}^{\Delta,h,\pi}(u) := \overline{r_{ij}(u)}^{\Delta,h,j} - \sigma_{ij}^h(\bar{u}^{\Delta,h,\pi}). \quad (98)$$

For comparison, the classical LES RST for the projected double-filter $\pi f^h f^\Delta$ would be $\tau_{\text{Classic},ij}^{\Delta,h,\pi} := \pi_{ij\alpha\beta}^h \xi_{\text{Classic},\alpha\beta}^{\Delta,h,\pi}$, where

$$\xi_{\text{Classic},ij}^{\Delta,h,\pi}(u) := \overline{\sigma_{ij}(u)}^{\Delta,h} - \sigma_{ij}^h(\bar{u}^{\Delta,h}). \quad (99)$$

Note that $\bar{u}^{\Delta,h}$ is already infinitesimally divergence-free, so $\pi_{ij} \bar{u}_j^{\Delta,h} = \bar{u}_i^{\Delta,h}$.

Like in the 1D case, the discretization-informed RST $\xi_{ij}^{\Delta,h,\pi}(u)$ accounts for both the filter-swap mechanism and the numerical stress σ_{ij}^h . Unlike the 1D case, it also accounts for the pressure, since the pressure that makes $\bar{u}^{\Delta,h,\pi}$ discretely divergence-free is not equal to a filtered variant of the pressure that makes u infinitesimally divergence-free.

The LES-FVM RST $\xi_{ij}^{\Delta,h,\pi}$ has two important differences from the classical LES RST $\xi_{\text{Classic}}^{\Delta,h,\pi}$ (classical LES with $\pi f^h f^\Delta$ as the filter). First, it is non-symmetric ($\xi_{ij}^{\Delta,h,\pi} \neq \xi_{ji}^{\Delta,h,\pi}$), while the classical RST is symmetric ($\xi_{\text{Classic},ij}^{\Delta,h,\pi} = \xi_{\text{Classic},ji}^{\Delta,h,\pi}$). Second, it is non-local in u (due to the pressure projector), while the classical RST is local in u .

5.6. LES-FVM Closure

Let $m^{\Delta,h,\pi}(\bar{u}^{\Delta,h,\pi}) \approx \text{dev}(\xi^{\Delta,h,\pi}(u))$ be a grid-compatible LES-FVM closure model for the deviatoric part of the RST defined as $\text{dev}(\sigma)_{ij} := \sigma_{ij} - \delta_{ij} \sigma_{kk}/3$ for all $\sigma \in U^{3 \times 3}$. Since $\partial_j^h \pi_{ij\alpha\beta}^h p \delta_{\alpha\beta} = 0$ for all scalar fields $p \in U$, we only need a closure model for the deviatoric part of the RST (the isotropic part will be absorbed by the pressure projector). The approximate LES-FVM equations are

$$L_i^h(v^{\Delta,h,\pi}) = -\partial_j^h \pi_{ij\alpha\beta}^h m_{\alpha\beta}^{\Delta,h,\pi}(v^{\Delta,h,\pi}), \quad (100)$$

where $v^{\Delta,h,\pi} \approx \bar{u}^{\Delta,h,\pi}$ is the approximate LES-FVM solution.

We say that the closure $m^{\Delta,h,\pi}$ is structural if it is chosen to approximate $\xi^{\Delta,h,\pi}$ directly, and functional if it is chosen to produce the same dissipation as $\xi^{\Delta,h,\pi}$. In 1D, an RST and its dissipation coefficient are fields of the same dimension (scalar field). The distinction between structural and functional models is therefore less clear in 1D. In 3D, the RST has 9 components, while the dissipation coefficient has 1 component (scalar field). There is therefore more freedom in how to choose a functional model than a structural model in 3D.

In the exact LES-FVM equation (for $\bar{u}^{\Delta,h,\pi}$), the RST is non-symmetric and non-local in u . This observation would suggest that the closure model in the approximate LES-FVM equation (for $v^{\Delta,h,\pi}$) should also be non-symmetric and non-local in $v^{\Delta,h,\pi}$. In the exact LES equation (for \bar{u}^Δ), the RST is both symmetric and local in u , which justifies using a symmetric and local closure model in classical LES (before discretization). This means that a “perfect” LES-FVM closure model needs to predict 9 different tensor components, while classical LES closures only need to predict 6 symmetric tensor components.

We now perform an experiment with the new RST expression.

6. Experiment: DNS-aided LES-FVM for 3D turbulence

Like for the Burgers equation, we employ a “DNS-aided LES-FVM” approach to assess the importance of the RST definition in the LES-FVM equations. We consider a 3D decaying turbulence test case in a periodic box. The equations are defined by the following

³ An alternative is therefore to solve for the *surface-averaged* field $\bar{u}_i^{\Delta,h,j}$ (no sum over i), since it becomes ∂_j^h -divergence-free after filtering ($\partial_j^h \bar{u}_i^{\Delta,h,j} = 0$) [16,36]. However, this field is not governed by a ∂_j^h -conservation law, since the surface-averaged momentum is not conserved discretely [16]. We will therefore not consider this option here.

unitless parameters. The domain size is $\ell := 1$. The viscosity is $\nu := 2 \times 10^{-4}$. The number of finite volumes in each of the 3 dimensions for DNS and LES are $N_{\text{DNS}} := 500$ and $N := 100$ respectively, with a compression factor of 5. The grid spacings are $h_{\text{DNS}} := \ell / N_{\text{DNS}}$ and $h := \ell / N$ (note that these are called h and H in [Appendix B](#), respectively). The LES filter f^Δ is a Gaussian of width $\Delta := 2h$. In the DNS discretization of the Gaussian kernel G^Δ , we include points up to 2 standard deviations out on each side (see [Appendix B](#) for details). The infinitesimal expressions from [Section 5](#) are all approximated on the DNS grid so that we can evaluate them based on the DNS solution, but we will still use the infinitesimal notation to avoid visual clutter.

6.1. Initialization

Let $u \in U^3$ be a velocity field and $\kappa \in \mathbb{N}$ be a scalar wavenumber. We define the energy spectrum of u at κ as

$$E(u, \kappa) := \frac{1}{2} \sum_{k \in K(\kappa)} \|\hat{u}(k)\|^2, \quad (101)$$

where $\hat{u}(k)$ is the Fourier transform of u at a wavenumber k and $K(\kappa) := \{k \in \mathbb{Z}^3 \mid \kappa \leq \|k\| < \kappa + 1\}$ is the shell of vector wavenumbers with magnitude between κ and $\kappa + 1$.

We initialize the solution u on the DNS grid through the following procedure, where \leftarrow denotes the assignment operator.

1. Sample a random field $u_i(x) \sim \mathcal{N}(0, 1)$ from a normal distribution for each $i \in \{1, 2, 3\}$ and each DNS grid point x . We do not define u_i outside the grid points.
2. Project the velocity: $u \leftarrow \pi u$.
3. Compute the discrete Fourier transform $\hat{u} \leftarrow \text{FFT}(u)$.
4. For all wavenumbers $\kappa \in \{0, 1, \dots, \lfloor \sqrt{3} N_{\text{DNS}}/2 \rfloor\}$, compute the current shell energy $E(u, \kappa)$, where $\lfloor \cdot \rfloor$ denotes the integer part. For $\kappa \geq N/2$, the shells are only partially filled, since the discrete Fourier transform gives a finite number of Fourier modes. Adjust the coefficients in the shell $K(\kappa)$ as

$$\hat{u}(k) \leftarrow \sqrt{\frac{P(\kappa)}{E(u, \kappa)}} \hat{u}(k), \quad \forall k \in K(\kappa), \quad (102)$$

where $P(\kappa)$ is a prescribed energy profile defined as the Kolmogorov scaling in the inertial range as

$$P(\kappa) := \kappa^{-5/3}. \quad (103)$$

5. Apply inverse Fourier transform $u \leftarrow \text{IFFT}(\hat{u})$.
6. Reproject the velocity field (since the shell normalization may slightly perturb the DNS divergence of u): $u \leftarrow \pi u$.
7. Scale the velocity field such that the total energy adds up to 1: $u \leftarrow \frac{1}{\sqrt{1/2\|u\|^2}} u$.

The resulting velocity field u is represented as an array of size $N_{\text{DNS}}^3 \times 3$. It is divergence-free, the spectrum is proportional to the profile P (with some deviations due to the second projection), and the total energy is 1.

Since the initial spectrum is artificial, we first run the DNS simulation for 0.1 time units to obtain a more realistic distribution of velocity scales.

6.2. DNS-Aided LES-FVM

The DNS-aided LES-FVM formulation is defined as

$$L(u) = 0, \quad L^h(v^{\text{model}}) = -\nabla^h \cdot \tau^{\text{model}}(u), \quad (104)$$

where u is the DNS solution, v^{model} is the modeled LES-FVM solution, $\nabla^h := (\partial_1^h, \partial_2^h, \partial_3^h)$, $\tau^{\text{model}} := \pi^h \xi^{\text{model}}$, and $\xi^{\text{model}} : U^3 \rightarrow U^{3 \times 3}$ is an RST expression taking the DNS solution as input (to be defined in [Section 6.3](#)). These equations are still continuous in time. A forward-Euler time discretization with adaptive time stepping is

$$\Delta t^n := C \times \min \left(\frac{h_{\text{DNS}}}{\max |u^n|}, \frac{h_{\text{DNS}}^2}{6\nu} \right), \quad (105)$$

$$u^{n+1} := u^n - \Delta t^n \nabla \cdot r(u^n), \quad (106)$$

$$v^{n+1} := v^n - \Delta t^n \nabla^h \cdot (r^h(v^n) + \tau^{\text{model}}(u^n)), \quad (107)$$

where Δt^n is chosen based on the DNS solution, u^n and v^n are the Forward-Euler approximations to the u and v^{model} at time $t^n := \sum_{k=0}^{n-1} \Delta t^k$, and $C := 0.4$ is a margin in the CFL expression. The initial conditions are given by the final DNS field from the warm-up simulation with $v^{\text{model}} = \bar{u}^{\Delta, h, \pi}$. The goal is for v^{model} to stay close to the projected filtered DNS velocity field $\bar{u}^{\Delta, h, \pi}$ over time.

Since both $u = \pi u$ and $v^h = \pi^h v^h$ are divergence-free on their respective grids, and [Theorem 8](#) can be used to commute projection and divergence, we can use the ‘‘semi-explicit’’ form of the scheme:

$$u^{n+1} = \pi [u^n - \Delta t^n \nabla \cdot \sigma(u)], \quad (108)$$

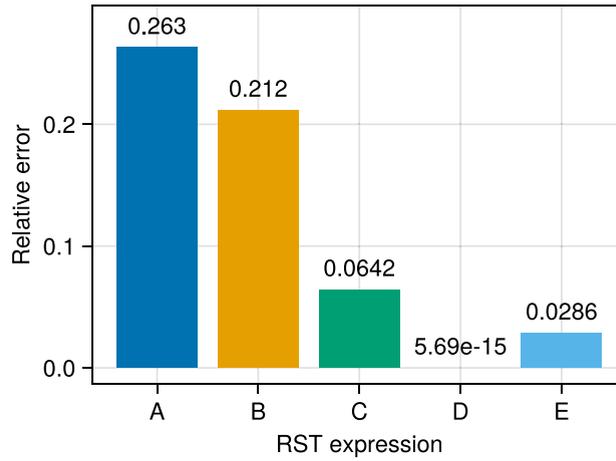


Fig. 12. Errors from DNS-aided LES-FVM for the 3D decaying turbulence simulation.

$$v^{n+1} = \pi^h \left[v^n - \Delta t^n \nabla^h \cdot \left(\sigma^h(v^n) + \xi^{\text{model}}(u^n) \right) \right], \quad (109)$$

which consists of doing an explicit forward-Euler step with the unprojected stresses before reprojecting the velocity field back onto the space of divergence-free fields.

Since turbulent flows are chaotic, we run the simulation for a short duration of 0.1 time units. If we run for longer, the LES and DNS solutions will decorrelate and the DNS-aided closure term will be of little use (unless ξ^{model} is the new correct RST expression).

6.3. RST Expressions

We consider five DNS-aided “closure” models defined (with no sum over j) by:

$$\xi_{ij}^A(u) := 0, \quad (110)$$

$$\xi_{ij}^B(u) := \overline{\sigma_{ij}(u)}^{\Delta,h} - \sigma_{ij}(\bar{u}^{\Delta,h}), \quad (111)$$

$$\xi_{ij}^C(u) := \overline{\sigma_{ij}(u)}^{\Delta,h} - \sigma_{ij}^h(\bar{u}^{\Delta,h,\pi}), \quad (112)$$

$$\xi_{ij}^D(u) := \overline{\sigma_{ij}(u)}^{\Delta,h,j} - \sigma_{ij}^h(\bar{u}^{\Delta,h,\pi}), \quad (113)$$

$$\xi_{ij}^E(u) := \frac{1}{2} \left(\xi_{ij}^D(u) + \xi_{ji}^D(u) \right). \quad (114)$$

The respective LES-FVM solutions are denoted $\{v^A, \dots, v^E\}$. ξ^A is the no-model case (DNS on the coarse grid). ξ^B is the classical RST expression for the given double-filter, with all operators evaluated on the DNS grid. Since $\pi f^{\Delta,h} = f^{\Delta,h}\pi$ and $\pi u = u$, the second term in $\xi_{ij}^B(u)$ simplifies from $\sigma(\pi \bar{u}^{\Delta,h})$ to $\sigma(\bar{u}^{\Delta,h})$. ξ^C accounts for the coarse grid discretization through the h -grid numerical stress σ^h in the second term. Since the filtered field is not divergence-free on the h -grid ($\pi^h f^{\Delta,h} \neq f^{\Delta,h}\pi$), the second term in ξ^C includes the projector π^h . ξ^D further accounts for the filter-swap mechanism. The FVM filter is therefore surface-averaging in the first term and not volume-averaging. ξ^E is defined as the symmetric part of ξ^D (the surface-averaged term makes ξ^D non-symmetric).

From Eq. (98), we know that ξ^D is the correct expression, since it accounts for the h -grid FVM discretization. In an “explicit LES”, which can be thought of as letting $h \rightarrow 0$, ξ^B would be the correct expression. ξ^C and ξ^E are included to assess the importance of the filter-swap mechanism and the non-symmetric part of the correct RST ξ^D , respectively. As classical LES closure models are designed to be symmetric, they are *at best* able to represent ξ^E , but not the non-symmetric RST ξ^D .

6.4. Results

In Fig. 12, we show the relative errors

$$\frac{\|v^{\text{model}} - \bar{u}^{\Delta,h,\pi}\|_{\Omega}}{\|\bar{u}^{\Delta,h,\pi}\|_{\Omega}} \quad (115)$$

at the final time $t = 0.1$ for the five RST expressions. Our proposed RST expression ξ^D gives an error at machine precision, confirming that it is the correct RST expression for the LES-FVM formulation. The error for ξ^E is at 2.86%, showing that the non-symmetric part of the RST is indeed non-zero. The error for ξ^C is at 6.42%, a bit more than twice as large as ξ^E . This shows that accounting for the filter-swap mechanism is also important. The error for the classical expression ξ^B is at 21.2%, a bit more than three times larger than for ξ^C . Finally, the no-model ξ^A gives the largest error at 26.3%. When comparing the errors for all the expressions, the largest absolute

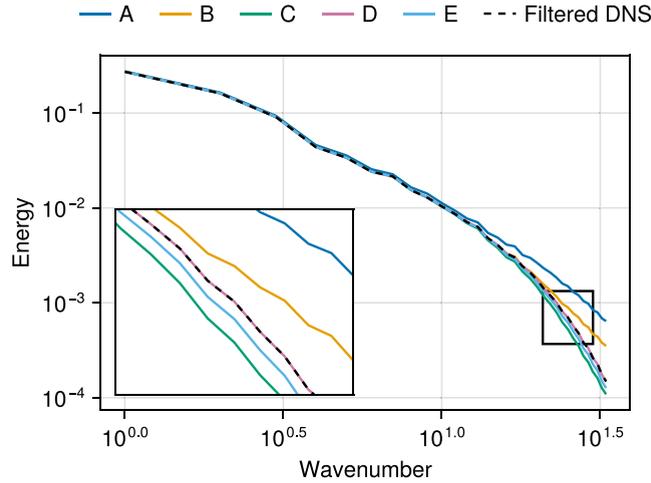


Fig. 13. Energy spectra from DNS-aided LES-FVM for the 3D decaying turbulence test case.

improvement is obtained by accounting for the numerical flux (from ξ^B to ξ^C). All errors are evaluated after a short simulation of 0.1 time units. They will therefore continue to grow for longer simulations.

In Fig. 13, we show the energy spectra at the final time. The spectrum of ξ^D perfectly overlaps with the reference spectrum, while the one of ξ^E is lower, and the one of ξ^C is lower still. This means that not accounting for the non-symmetry and the filter-swap property leads to a slight over-estimation of the energy dissipation. On the other hand, the spectrum of the classical RST ξ^B is too high, meaning that not accounting for the discretization in the RST leads to an under-estimation of the energy dissipation. Finally, the no-model ξ^A has the highest spectrum, showing that a dissipative closure model is required to get the correct spectrum.

In the 1D Burgers' case (Fig. 7), many of the residual flux expressions led to a sharp increase at the very end of the spectrum. This was likely due to the shock-like structures that cannot be resolved on the coarse grid. In Fig. 13, no such sharp increase at the end of the spectrum is observed. Although the 3D resolution is smaller, which makes it difficult to compare with the 1D case, the 3D experiment is incompressible, and therefore cannot form the shock-like structures that were present in the Burgers' case.

7. Conclusion

In this work, we have proposed new exact expressions for the residual stress tensor (RST) appearing in discrete filtered conservation laws when using combined LES and FVM filters. Unlike the classical RST expression $\overline{uu} - \overline{u}u$ which is commonly studied in LES, the RST in LES-FVM also includes contributions from the discrete divergence, numerical flux, and discrete incompressibility constraint. Writing the residual in discrete divergence form makes the RST non-symmetric, while accounting for the discrete incompressibility constraint makes the RST non-local. To make these two properties more intuitive, we show how they are derived using an alternative notation in Appendix E.

In a DNS-aided LES framework, where DNS data is used to compute the RST, our RST expression gives errors at machine precision, while the classical RST gives errors that grow over time, confirming the importance of accounting for the discretization errors in the RST.

In practice, DNS-aided LES is only useful for evaluating the correctness of the RST expression. Does the new RST provide any practical benefits for LES closure modeling? We believe so, and we have emphasized two main benefits:

1. The new framework informs us about the structure of the RST, which can guide the development of new or existing closure models. Notably, the LES-FVM RST expression suggests that a "perfect" structural closure model for incompressible flows should be non-symmetric and non-local in the velocity field. From the experiments in Section 6, we have some idea of the importance of the non-symmetric part, but the benefit of including non-local values in a closure model for incompressible flows is still unclear and requires further study.
2. The new RST can be used as a target for tuning closure model parameters. By correctly accounting for the discretization in the RST, the RST automatically gives the correct desired dissipation profile that a functional closure model should reproduce. Notably, the correct RST accounts for the implicit dissipation produced by the numerical flux, so that the amount of explicitly modeled dissipation can be reduced accordingly. Our LES-FVM framework can thus be used as a tool to bridge the gap between implicit and explicit LES closure modeling.

For the 1D Burgers equation, we demonstrated the benefit of the second point by fitting the Smagorinsky model coefficient to the old and new RST expression. Indeed, the coefficient fitted to the new RST was slightly higher, adding some of the additional dissipation required by the FVM discretization and leading to better results. This is despite the fact that the Smagorinsky model only has one coefficient and is a rather inaccurate closure.

To compute the RST exactly from DNS reference data, we have introduced a new *two-grid formulation* that accounts for two different overlapping discretization levels (see [Appendix B](#)). This formulation is constructed such that the filter-swap commutation properties hold discretely. Making sure the DNS and LES grids are compatible in this way ensures that the exact RST (with respect to the DNS reference) is computable. Furthermore, as high-dimensional function approximators such as neural networks are receiving more attention for LES closure modeling [37], access to accurate discretization-informed target data becomes increasingly important [7].

For 1D conservation laws, the central insight that leads to these discrete formulations was the partial restoration of commutation between filtering and differentiation for finite differences and coarsening grid filters. We showed that coarsening filters do not commute with differentiation, but the partial commutation property of finite differences was sufficient to obtain a discrete RST expression. This commutation property also holds for higher-order finite difference schemes, such as the ones studied by Geurts and van der Bos [11]. A similar formulation can be derived for finite volume schemes on unstructured grids, since a volume-averaged divergence over an unstructured grid cell can be written as a surface integral over the volume faces. Such an RST expression was proposed by Denaro [24,25].

The key to making the LES-FVM framework valid for incompressible flows was to rewrite the incompressible Navier-Stokes equations as a self-contained conservation law for the velocity field. This was possible by incorporating the incompressibility constraint and pressure gradient correction into a pressure projection operator. This operator then became part of the stress tensor in the self-contained conservation law for the velocity field. In the presence of no-slip boundary conditions, this approach can still be used. However, it is not yet clear to us if the pressure term can be written solely in terms of the velocity for certain types of outflow boundary conditions that prescribe a value for the pressure at the boundary. This topic requires further study.

The LES-FVM framework we propose relies on first choosing a discretization and a grid size, before choosing a closure model and tuning its parameters. We therefore cannot expect the closure model to work well for a different discretization or a different grid size. The discrete closure model must be recalibrated to discretization-informed target data obtained from DNS when the LES grid size is changed.

We demonstrated the LES-FVM framework for staggered grids, which are composed of a primal and a dual grid (velocity and pressure points). The nodes on the primal and dual grids do not overlap but are separated by half a grid spacing. This led to the requirement that the primal and dual LES grids overlap with the primal and dual DNS grids, respectively, by using uniform Cartesian grids with odd compression factors. This is a limitation of our framework on staggered grids. With some care, *unstructured* staggered grids can also be designed such that the primal and dual LES grids overlap with the primal and dual DNS grids. This can be achieved by *first* choosing the LES grid and then refining it to obtain the DNS for generating training data. For flows around objects (such as airfoils), the LES grid would therefore need to be sufficiently fine to resolve the shape of the object considered. No further refinement of the object boundary would be possible without losing the exactness of the formulation. By allowing for some error at the boundary, this requirement could be relaxed, while retaining the exact formulation in the interior of the domain.

Software and reproducibility statement

The code used to generate the results of this paper is available at <https://github.com/agdestein/ExactClosure.jl>. It is released under the MIT license. An archived version is available at <https://zenodo.org/records/16267799>. The results in [Section 4](#) were obtained on a desktop CPU. The results in [Section 6](#) were obtained using a single Nvidia H100 GPU on the Dutch National Supercomputer Snellius.

To allow for reproducibility on accessible hardware, the code also includes a scaled down version of the 3D experiment (90^3 DNS grid points and $\{18^3, 30^3\}$ LES grid points) that can be run with multithreading on a modern laptop CPU. The results for the scaled-down 3D experiment are similar to those of the full experiment.

CRedit authorship contribution statement

Syver Døving Agdestein: Writing – original draft, Visualization, Validation, Software, Methodology, Conceptualization; **Roel Verstappen:** Writing – review & editing, Supervision; **Benjamin Sanderse:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Formal analysis.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used GitHub Copilot in order to propose wordings and mathematical typesetting. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Data availability

The code is open source and linked in the article.

Declaration of competing interest

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

Acknowledgements

This work is supported by the projects ‘‘Discretize first, reduce next’’ (with project number VI.Vidi.193.105) of the research programme NWO Talent Programme Vidi and ‘‘Discovering deep physics models with differentiable programming’’ (with project number EINF-8705) of the Dutch Collaborating University Computing Facilities (SURF), both financed by the Dutch Research Council (NWO). We thank SURF (<http://www.surf.nl>) for the support in using the Dutch National Supercomputer Snellius.

Appendix A. Grid-compatible operators and restriction

We say that an operator $o : U \rightarrow U$ is h -collocated if, for all $x \in \Omega$, $o(u)(x)$ only depends on the values $\{u(x + ih) \mid i \in \mathbb{Z}\}$. Similarly, o is h -staggered if $o(u)(x)$ only depends on $\{u(x + ih + h/2) \mid i \in \mathbb{Z}\}$.

The finite difference ∂_x^h , interpolator η_x^h , and numerical Burgers flux r^h from Eqs. (13), (14), and (15) are staggered since $\partial_x^h u(x)$, $\eta_x^h u(x)$, and $r^h(u)(x)$ depend on $u(x \pm h/2)$, and not on $u(x \pm h)$. When chained together, the staggered operators become collocated. For example, the expressions

$$\partial_x^h \partial_x^h u(x) = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}, \quad (\text{A.1})$$

$$\partial_x^h \eta_x^h u(x) = \frac{u(x+h) - u(x-h)}{2h}, \quad (\text{A.2})$$

only depend on $u(x-h)$, $u(x)$, and $u(x+h)$. These expressions do not depend on u at the half-points $x \pm h/2$.

We say that an operator (such as the discrete Laplacian $\partial_x^h \partial_x^h$) is *compatible* with a grid of spacing h if the operator is collocated. This means that equations involving the operator form a closed system of equations once restricted to the grid points. In particular, a flux term of the form $\partial_x^h r^h$ is grid-compatible if $r^h : U \rightarrow U$ is a staggered numerical flux. We therefore say that a flux is grid-compatible if it is staggered.

At a point x , the numerical flux Burgers flux $r^h(u) := (\eta_x^h u)(\eta_x^h u)/2 - \nu \partial_x^h u$ takes the form

$$\begin{aligned} r^h(u)(x) &= \frac{1}{8} \left[u\left(x - \frac{h}{2}\right) + u\left(x + \frac{h}{2}\right) \right]^2 \\ &\quad - \frac{\nu}{h} \left[u\left(x + \frac{h}{2}\right) - u\left(x - \frac{h}{2}\right) \right], \end{aligned} \quad (\text{A.3})$$

and the discrete flux divergence is

$$\begin{aligned} \partial_x^h r^h(u)(x) &= \frac{1}{h} \left[r^h(u)\left(x + \frac{h}{2}\right) - r^h(u)\left(x - \frac{h}{2}\right) \right] \\ &= \frac{1}{8h} \left[u(x+h)^2 - u(x-h)^2 \right] \\ &\quad + \frac{1}{8h} u(x) [u(x+h) - u(x-h)] \\ &\quad - \frac{\nu}{h^2} [u(x+h) - 2u(x) + u(x-h)]. \end{aligned} \quad (\text{A.4})$$

This expression can be evaluated at any point $x \in \Omega$. If we evaluate the continuous field $\partial_x^h r^h(u)$ at the grid points $x_i^h := ih$, $i \in \mathbb{Z}$, and define the restriction $u_i^h := u(x_i^h)$, we can use the more common (but less general) notation style

$$\begin{aligned} \partial_x^h r^h(u)(x_i^h) &= \frac{1}{8h} \left[(u_{i+1}^h)^2 - (u_{i-1}^h)^2 + u_i^h (u_{i+1}^h - u_{i-1}^h) \right] \\ &\quad - \frac{\nu}{h^2} [u_{i+1}^h - 2u_i^h + u_{i-1}^h]. \end{aligned} \quad (\text{A.5})$$

This restricted form, where the expression is written only in terms of $(u_i^h)_{i \in \mathbb{Z}}$, is only possible to write because $\partial_x^h r^h$ is a grid-compatible operator. We still use the continuous notation like in Eqs. (15), (A.3), and (A.4) for analysis and only use the restricted form like (A.5) for computer evaluation.

Appendix B. Grid-compatible coarse-graining

A problem with the LES-FVM framework is that we cannot compute target solutions $\bar{u}^{\Delta,h}$ and target residual fluxes $\tau^{\Delta,h}(u)$ unless we have access to a continuous solution u in all points $x \in \Omega$. In practice, reference data is obtained by first doing DNS by solving the DNS equation

$$L^h(u^h) = 0 \quad (\text{B.1})$$

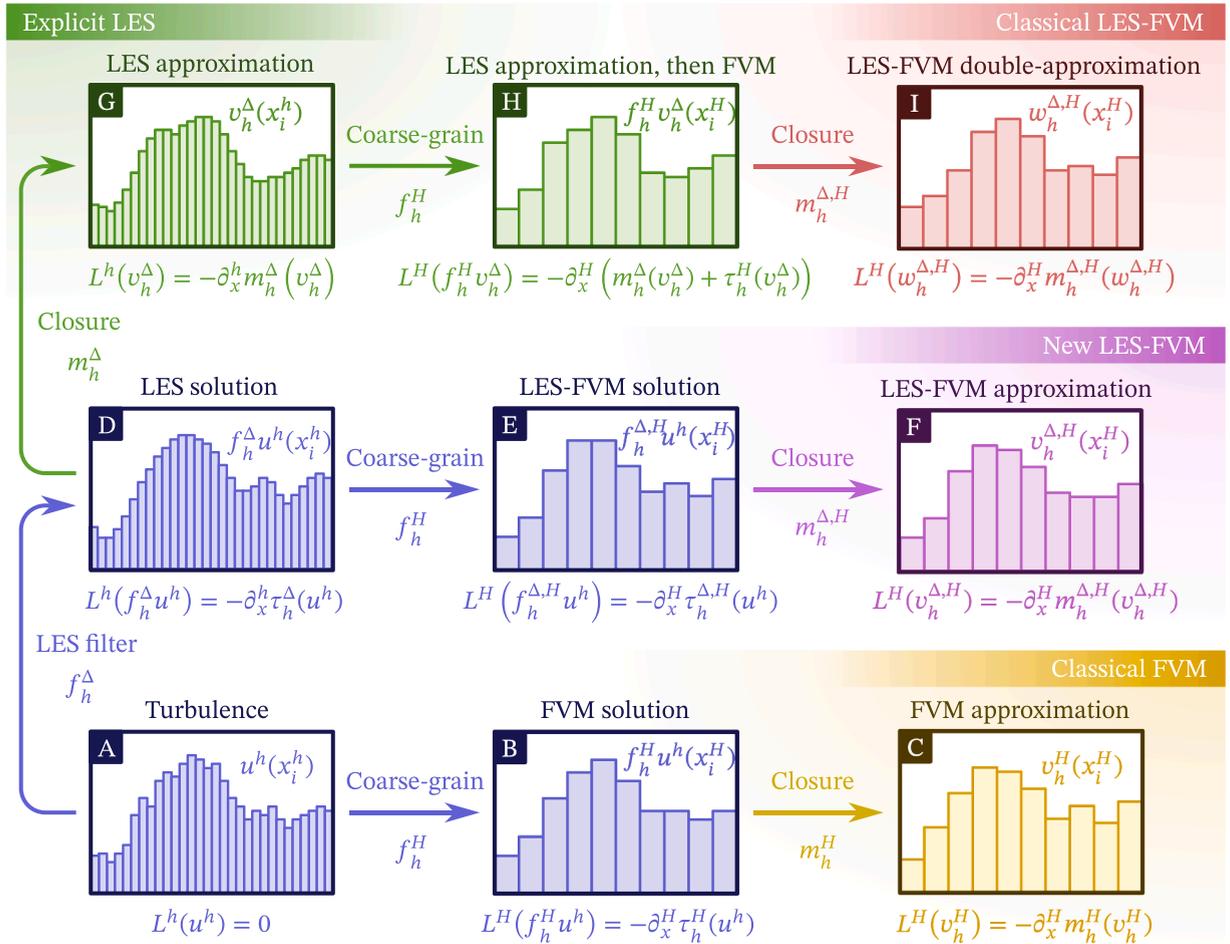


Fig. B.14. Same as Fig. 1, but starting from the DNS solution u^h instead of the continuous solution u . In the limit when h goes to zero, this figure becomes equivalent to Fig. 1. The fine grid points are $(x_i^h)_{i \in \mathbb{Z}}$, and the coarse grid points are $(x_i^H)_{i \in \mathbb{Z}}$.

with a sufficiently small DNS grid spacing h . This gives the DNS solution u^h at the h -grid points $x_i^h := ih, i \in \mathbb{Z}$. The LES-FVM problem is then formulated on a coarser grid with spacing $H > h$. The goal is to approximate the fields $\bar{u}^{\Delta,H}(x_i^H)$ and $\tau^{\Delta,H}(u)(x_i^H)$ evaluated at the H -grid points $(x_i^H)_{i \in \mathbb{Z}}$ using only the DNS solution $u^h(x_i^h)$ evaluated at the h -grid points $(x_i^h)_{i \in \mathbb{Z}}$. This framework is illustrated in Fig. B.14. This two-grid formulation should approximate the one-grid formulation and converge to it as h goes to 0.

B.1. 1D Filters

For the 1D case, we propose the following DNS approximations of the filters that fits our criteria. Choose $H := (2n + 1)h$ for some $n \in \mathbb{N}$. The FVM filter f^H is replaced by the h -compatible filter $f_h^H : U \rightarrow U$ using the quadrature rule for all $u \in U$ and $x \in \Omega$ as

$$f_h^H u(x) := \frac{1}{2n + 1} \sum_{i=-n}^n u(x + ih). \tag{B.2}$$

This filter is designed to satisfy a discrete equivalent of the filter-swap commutation property (19):

$$\partial_x^H = f_h^H \partial_x^h. \tag{B.3}$$

For proof, see Theorem 5.

Assume that the LES kernel G^Δ of the LES filter f^Δ is compactly supported (e.g. top-hat) or decays sufficiently fast at infinity (e.g. Gaussian). We then replace f^Δ with the h -compatible filter $f_h^\Delta : U \rightarrow U$ defined as

$$f_h^\Delta u(x) = \frac{\sum_{r=-R}^R G^\Delta(rh)u(x - rh)}{\sum_{r=-R}^R G^\Delta(rh)} \tag{B.4}$$

for some sufficiently large $R \in \mathbb{N}$. The denominator ensures that the discrete kernel is normalized. For the Gaussian filter (38), the standard deviation is $\sigma = \Delta/\sqrt{12}$. We then choose R such that $Rh \geq 3\sigma$, i.e. $R := \lceil 3\Delta/(\sqrt{12}h) \rceil$, where $\lceil \cdot \rceil$ is the ceiling function. The filter f_h^Δ is h -compatible since $f_h^\Delta u(x)$ only depend on u at x modulo h .

Applying the coarse-graining h -compatible LES-FVM filter $f_h^{\Delta,H} := f_h^H f_h^\Delta$ to the DNS Eq. (B.1) gives the LES-FVM equation

$$f_h^{\Delta,H} L^h(u^h) = 0. \quad (\text{B.5})$$

By adding the resolved term $\partial_x^H r^H(f_h^{\Delta,H} u^h)$ to both sides and using the discrete filter-swap property (B.3), we get the LES-FVM equation in H -grid conservative form:

$$L^H(f_h^{\Delta,H} u^h) = -\partial_x^H \tau_h^{\Delta,H}(u^h) \quad (\text{B.6})$$

(see Fig. B.14-E), where $f_h^{\Delta,H} u^h$ is the coarse-grained LES-FVM solution we intend to solve for and

$$\tau_h^{\Delta,H}(u^h) := f_h^\Delta r^h(u^h) - r^H(f_h^{\Delta,H} u^h) \quad (\text{B.7})$$

is the residual flux in the LES-FVM equation. Eq. (B.6) is analogous to the LES-FVM Eq. (34), but it is written using H -grid divergences ∂_x^H .

Note that the FVM filters f^h , f^H , and f_h^H are related through the property

$$f^H = f_h^H f^h. \quad (\text{B.8})$$

For proof, see Theorem 4. At fixed H , we can show that $f_h^H \rightarrow f^H$ and $f_h^\Delta \rightarrow f^\Delta$ as $h \rightarrow 0$. This means that if the DNS is fully resolved, we recover the continuous setting.

A limitation of our FVM filter f_h^H is that the compression factor is required to be odd, i.e. $H = (2n+1)h$ for some n , and not $H = 2nh$. For an odd compression factor, we can compute both $f_h^{\Delta,H} u^h$ and $\tau_h^{\Delta,H}(u^h)$ in the required staggered grid points exactly, without performing any interpolations. This would not be possible for an even compression factor, due to the way the staggered coarse and fine grids overlap.

B.2. Coarse-graining filters in 3D

The 1D h -compatible LES and FVM filters can easily be extended to 3D by applying them successively in each coordinate direction. The 1D FVM filter $g_{h,i}^H : U \rightarrow U$ can be defined as

$$g_{h,i}^H u(x) := \frac{1}{2n+1} \sum_{r=-n}^n u(x + r h e_i), \quad (\text{B.9})$$

where e_i is the i -th unit vector. The volume-averaging and surface-averaging FVM filters then follow as $f_h^H := g_{h,1}^H g_{h,2}^H g_{h,3}^H$, $f_h^{H,1} := g_{h,2}^H g_{h,3}^H$, $f_h^{H,2} := g_{h,1}^H g_{h,3}^H$, and $f_h^{H,3} := g_{h,1}^H g_{h,2}^H$.

Like their one-grid counterparts, our proposed two-grid filters have the commutation properties (with no sum over i)

$$g_{h,i}^H \partial_i^h = \partial_i^H, \quad (\text{B.10})$$

$$f_h^H \partial_i^h = \partial_i^H f_h^{H,i}. \quad (\text{B.11})$$

Given a DNS solution u^h restricted to the h -grid, target data pairs $f_h^{\Delta,H,\pi} u^h$ and $\tau_h^{\Delta,H,\pi}(u^h)$ can be computed and then evaluated in the required H -grid points to assess the performance of LES-FVM closures.

Appendix C. Proofs of commutation properties

Here we provide proofs for various properties used in this article. We recall that Ω is a periodic 1D domain, U is the space of periodic 1D fields on Ω , f^Δ is a spatial convolutional LES filter (see Eq. (5)), f^h is an FVM filter (see Eq. (17)), f_h^Δ is a h -grid-compatible LES filter (see Eq. (B.2)), f_h^H is a h -grid-compatible FVM filter (see Eq. (B.2)), ∂_x^H is a finite difference operator (see Eq. (13)), h is a grid spacing, and $H = (2n+1)h$ is a coarse grid spacing for some $n \in \mathbb{N}$.

Note that for non-uniform filters or bounded domains, some of the commutation properties may no longer hold. Here, we only consider periodic domains.

Theorem 1. *Spatial convolutional filters f^Δ commute with differentiation [2]:*

$$f^\Delta \partial_x = \partial_x f^\Delta. \quad (\text{C.1})$$

Proof. Let $u \in U$ and $x \in \Omega$. Then

$$\begin{aligned} \partial_x \bar{u}^\Delta(x) &= \partial_x \left[\int_{\mathbb{R}} G^\Delta(x-y)u(y) dy \right] \\ &= \int_{\mathbb{R}} \partial_x [G^\Delta(x-y)]u(y) dy \\ &= \int_{\mathbb{R}} (\partial_x G^\Delta)(x-y)u(y) dy \\ &= - \int_{\mathbb{R}} \partial_y [G^\Delta(x-y)]u(y) dy \\ &= \int_{\mathbb{R}} G^\Delta(x-y)\partial_y u(y) dy \\ &= \overline{\partial_x u}^\Delta(x). \end{aligned} \tag{C.2}$$

where we first used Leibnitz’s rule to interchange differentiation and integration, then the chain rule, and then integration by parts (assuming that the kernel goes to zero at infinity). Since this holds for all u and x , we have $f \partial_x = \partial_x f$. \square

Theorem 2. Coarse-graining and differentiation do not commute:

$$\partial_x^h f^h \neq f^h \partial_x. \tag{C.3}$$

Proof. Let $u \in U$. The Taylor series expansion of u around a point x is

$$\begin{aligned} u\left(x + \frac{h}{2}\right) &= u(x) + \frac{h}{2}\partial_x u(x) + \frac{h^2}{8}\partial_{xx}u(x) \\ &\quad + \frac{h^3}{48}\partial_{xxx}u(x) + \frac{h^4}{384}\partial_{xxxx}u(x) + \mathcal{O}(h^5), \end{aligned} \tag{C.4}$$

where $\partial_{xx} := \partial_x \partial_x$ etc. Subtracting a similar expansion of $u(x - h/2)$ makes the even terms cancel out. The expansion of the finite difference operator ∂_x^h is therefore reduced to

$$\partial_x^h = \partial_x - \frac{h^2}{24}\partial_{xxx} + \mathcal{O}(h^4). \tag{C.5}$$

This gives

$$\begin{aligned} \partial_x^h f^h &= \partial_x f^h - \frac{h^2}{24}\partial_{xxx} f^h + \mathcal{O}(h^4) \\ &= f^h \partial_x - \frac{h^2}{24} f^h \partial_{xxx} + \mathcal{O}(h^4) \\ &\neq f^h \partial_x, \end{aligned} \tag{C.6}$$

since in the operator $f^h \partial_{xxx}$ is non-zero. Here we used **Theorem 1** to swap f^h and ∂_x . Note that for a few special cases, such as velocity fields with $\partial_{xxx}u = 0$ (and similarly for higher order derivatives), we do get $\partial_x^h \bar{u}^h = \overline{\partial_x u}^h$. \square

Theorem 3. The finite difference ∂_x^h can be written as a composition between the FVM filter and an exact derivative:

$$\partial_x^h = f^h \partial_x. \tag{C.7}$$

Proof. The fundamental theorem of calculus states that

$$\int_a^b \partial_x u dx = u(b) - u(a) \tag{C.8}$$

for all $(a, b) \in \mathbb{R}^2$. For $u \in U$ and $x \in \Omega$, this gives

$$\begin{aligned} \overline{\partial_x u}^h(x) &= \frac{1}{h} \int_{x-h/2}^{x+h/2} \partial_y u(y) dy \\ &= \frac{1}{h} \left[u\left(x + \frac{h}{2}\right) - u\left(x - \frac{h}{2}\right) \right] \\ &= \partial_x^h u(x). \end{aligned} \tag{C.9}$$

Since this holds for all u and x , we have $\partial_x^h = f^h \partial_x$. \square

We now show the properties of the coarse-graining filter f_h^H .

Theorem 4. The average over $H := (2n + 1)h$ can be written as a composition between a coarse-graining filter and the average over h :

$$f^H = f_h^H f^h, \tag{C.10}$$

where $n \in \mathbb{N}$.

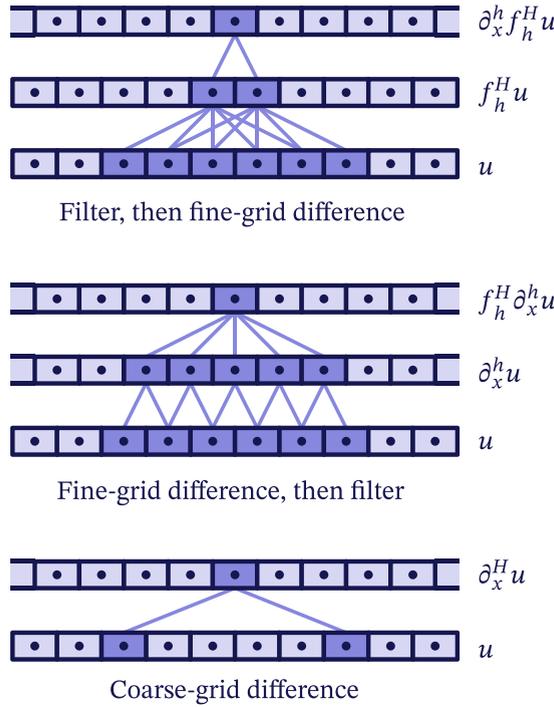


Fig. C.15. For all $u \in U$, the three terms $\partial_x^H u$, $f_h^H \partial_x^h u$, and $\partial_x^h f_h^H u$ are equal. Here we show the three quantities on the h -grid with $H := 5h$.

Proof. Let $u \in U$ and $x \in \Omega$. Then

$$\begin{aligned}
 f_h^H \bar{u}^h(x) &= \frac{1}{2n+1} \sum_{i=-n}^n \frac{1}{h} \int_{x+ih-h/2}^{x+ih+h/2} u(y) \, dy \\
 &= \frac{1}{(2n+1)h} \int_{x-(2n+1)h/2}^{x+(2n+1)h/2} u(y) \, dy \\
 &= \frac{1}{H} \int_{x-H/2}^{x+H/2} u(y) \, dy \\
 &= \bar{u}^H(x),
 \end{aligned}
 \tag{C.11}$$

where we used the property $\int_a^b u(x)dx + \int_b^c u(x)dx = \int_a^c u(x)dx$ for all $(a, b, c) \in \mathbb{R}^3$ to combine the integrals in the sum. Since this holds for all u and x , we have $f^H = f^{h \rightarrow H} f^h$. \square

Theorem 5. A finite difference over $H := (2n + 1)h$ can be written as a composition between a coarse-graining filter and a finite difference over h :

$$\partial_x^H = f_h^H \partial_x^h,
 \tag{C.12}$$

where $n \in \mathbb{N}$.

Proof. Using Theorems 3 and 4, we have

$$\partial_x^H = f^H \partial_x = f_h^H f^h \partial_x = f_h^H \partial_x^h.
 \tag{C.13}$$

\square

Note that Theorem 5 can be extended to include three terms: $\partial_x^H = \partial_x^h f_h^H = f_h^H \partial_x^h$, since filtering and finite differencing do commute if we do not coarse-grain the derivative. These three terms are shown in Fig. C.15 for $H = 5h$. The three terms are equal since all the intermediate terms cancel out in the telescoping sum in f_h^H . We restrict the fields to the h -grid to visualize how the inner terms cancel out.

Appendix D. Pressure projection for vectors and stress tensors

The system (70) consist of an evolution equation subject to a spatial constraint. By defining a pressure projection operator, these equations can be combined into one self-contained evolution equation.

In this appendix, we introduce two projection operators, one that makes vector fields in U^3 divergence-free and one that makes stress tensor fields in $U^{3 \times 3}$ divergence-preserving. We provide proofs for the continuous case. For the discrete case, the proofs are the identical, the only difference is that instead of ∂_i we use ∂_i^h .

D.1. Pressure projector for vector fields

The pressure field p enforces the continuity equation $\partial_j u_j = 0$. In Eq. (70), combining the continuity equation with the momentum equation gives the Poisson equation for the pressure field:

$$-\partial_k \partial_k p = \partial_i \partial_j \sigma_{ij}(u). \quad (\text{D.1})$$

By solving this equation explicitly for the pressure, we can write a ‘‘pressure-free’’ momentum equation as

$$\partial_i u_i + \pi_{ij} \partial_k \sigma_{jk}(u) = 0, \quad (\text{D.2})$$

where

$$\pi_{ij} := \delta_{ij} - \partial_i (\partial_k \partial_k)^{\dagger} \partial_j \quad (\text{D.3})$$

is a pressure projection operator (see Theorem 6 for proof), δ_{ij} is the Kronecker symbol, and the inverse Laplacian $(\partial_k \partial_k)^{\dagger} : \varphi \mapsto p$ maps scalar fields φ to the unique solution to the Poisson equation $\partial_k \partial_k p = \varphi$ subject to the additional constraint of an average pressure of zero, i.e. $\int_{\Omega} p dV = 0$. For our periodic domain, the pressure field is determined up to a constant. We are free to choose the constant this way since it subsequently disappears in the pressure gradient $\partial_i p$.

Theorem 6. *The operator $\pi_{ij} := \delta_{ij} - \partial_i (\partial_k \partial_k)^{\dagger} \partial_j$ is a projector onto the space of divergence-free vector fields, i.e. $\partial_i \pi_{ij} = 0$ (the output of π is divergence-free) and $\pi \pi = \pi$ [38–40].*

Proof. We have $\partial_i \partial_i (\partial_j \partial_j)^{\dagger} = 1$, since $(\partial_j \partial_j)^{\dagger}$ gives solutions to the Poisson equation. This can be used to show that π_{ij} makes vector fields divergence-free. The divergence ∂_i composed with π_{ij} is

$$\partial_i \pi_{ij} = \underbrace{\partial_i \delta_{ij}}_{\partial_j} - \underbrace{\partial_i \partial_i (\partial_k \partial_k)^{\dagger}}_1 \partial_j = \partial_j - \partial_j = 0. \quad (\text{D.4})$$

This means that for all $u \in U^3$, $\partial_i \pi_{ij} u_j = 0$, so πu is divergence-free (even if u is not). Additionally, we get

$$\begin{aligned} \pi_{ij} \pi_{jk} &= \delta_{ij} \delta_{jk} \\ &\quad - \delta_{ij} \partial_j (\partial_{\alpha} \partial_{\alpha})^{\dagger} \partial_k - \delta_{jk} \partial_i (\partial_{\alpha} \partial_{\alpha})^{\dagger} \partial_j \\ &\quad + \partial_i (\partial_{\alpha} \partial_{\alpha})^{\dagger} \partial_j \partial_j (\partial_{\beta} \partial_{\beta})^{\dagger} \partial_k \\ &= \delta_{ik} - 2 \partial_i (\partial_{\alpha} \partial_{\alpha})^{\dagger} \partial_k + \partial_i (\partial_{\alpha} \partial_{\alpha})^{\dagger} \partial_k \\ &= \pi_{ik}. \end{aligned} \quad (\text{D.5})$$

Since $\pi \pi = \pi$, we can conclude that π is idempotent. \square

The projected momentum Eq. (D.2) automatically enforces the continuity equation at all times by construction (as long as $\partial_j u_j = 0$ at the initial time). The pressure term and continuity equation can be ignored, at the cost of making the momentum equations non-local (the inverse Laplacian is non-local).

D.2. Pressure projector for tensor fields

We say that a stress tensor $\sigma \in U^{3 \times 3}$ is divergence-preserving if $\partial_j \sigma_{ij}$ is divergence-free, i.e. $\partial_i \partial_j \sigma_{ij} = 0$. Define the tensor projector $\pi : U^{3 \times 3} \rightarrow U^{3 \times 3}$ as

$$\pi_{ij\alpha\beta} := \delta_{i\alpha} \delta_{j\beta} - \delta_{ij} (\partial_k \partial_k)^{\dagger} \partial_{\alpha} \partial_{\beta}. \quad (\text{D.6})$$

This operator maps stress tensors to stress tensors.

Theorem 7. *The operator $\pi_{ij\alpha\beta}$ is a projector onto the space of divergence-preserving stress tensors, i.e. $\partial_i \partial_j \pi_{ij\alpha\beta} = 0$ (the output of π is divergence-preserving) and $\pi \pi = \pi$.*

Proof. The double-divergence $\partial_i \partial_j$ composed with the operator $\pi_{ij\alpha\beta}$ is

$$\begin{aligned} \partial_i \partial_j \pi_{ij\alpha\beta} &= \delta_{i\alpha} \delta_{j\beta} \partial_i \partial_j - \delta_{ij} \partial_i \partial_j (\partial_k \partial_k)^{\dagger} \partial_{\alpha} \partial_{\beta} \\ &= \delta_{i\alpha} \delta_{j\beta} - \underbrace{\delta_{ij} \partial_i \partial_j (\partial_k \partial_k)^{\dagger}}_1 \partial_{\alpha} \partial_{\beta} \\ &= \delta_{i\alpha} \delta_{j\beta} - \delta_{i\alpha} \delta_{j\beta} \\ &= 0. \end{aligned} \quad (\text{D.7})$$

This means that for all $\sigma \in U^{3 \times 3}$, we have $\partial_i \partial_j (\pi_{ij\alpha\beta} \sigma_{\alpha\beta}) = 0$, so $\pi\sigma$ is a divergence-preserving tensor.

Furthermore, applying the operator twice gives

$$\begin{aligned}
\pi_{ij\alpha\beta} \pi_{\alpha\beta mn} &= (\delta_{i\alpha} \delta_{j\beta} - \delta_{ij} (\partial_k \partial_k)^\dagger \partial_\alpha \partial_\beta) \\
&\quad (\delta_{\alpha m} \delta_{\beta n} - \delta_{\alpha\beta} (\partial_l \partial_l)^\dagger \partial_m \partial_n) \\
&= (\delta_{i\alpha} \delta_{j\beta}) (\delta_{\alpha m} \delta_{\beta n}) \\
&\quad - (\delta_{i\alpha} \delta_{j\beta}) \delta_{\alpha\beta} (\partial_l \partial_l)^\dagger \partial_m \partial_n \\
&\quad - (\delta_{\alpha m} \delta_{\beta n}) \delta_{ij} (\partial_k \partial_k)^\dagger \partial_\alpha \partial_\beta \\
&\quad + \delta_{ij} \delta_{\alpha\beta} (\partial_k \partial_k)^\dagger \partial_\alpha \partial_\beta (\partial_l \partial_l)^\dagger \partial_m \partial_n \\
&= \delta_{im} \delta_{jn} \\
&\quad - \delta_{ij} (\partial_l \partial_l)^\dagger \partial_m \partial_n \\
&\quad - \delta_{ij} (\partial_k \partial_k)^\dagger \partial_m \partial_n \\
&\quad + \delta_{ij} (\partial_k \partial_k)^\dagger \underbrace{\partial_\alpha \partial_\alpha (\partial_l \partial_l)^\dagger}_1 \partial_m \partial_n \\
&= \delta_{im} \delta_{jn} + (-2 + 1) \delta_{ij} (\partial_k \partial_k)^\dagger \partial_m \partial_n \\
&= \pi_{ijmn}.
\end{aligned} \tag{D.8}$$

Since $\pi\pi = \pi$, we can conclude that π is idempotent. \square

The vector-projector π_{ij} and tensor-projector $\pi_{ij\alpha\beta}$ satisfy the following commutation property for the tensor-divergence.

Theorem 8. *Projection and tensor-divergence commute, i.e. for all stress tensors $\sigma \in U^{3 \times 3}$, we have*

$$\pi_{ij} \partial_k \sigma_{jk} = \partial_j \pi_{ij\alpha\beta} \sigma_{\alpha\beta}. \tag{D.9}$$

Proof. Let $\sigma \in U^{3 \times 3}$ be a stress tensor. The tensor-divergence ∂_j of the projected stress tensor $\pi_{ij\alpha\beta} \sigma_{\alpha\beta}$ is

$$\begin{aligned}
\partial_j \pi_{ij\alpha\beta} \sigma_{\alpha\beta} &= \partial_j (\delta_{i\alpha} \delta_{j\beta} - \delta_{ij} (\partial_k \partial_k)^\dagger \partial_\alpha \partial_\beta) \sigma_{\alpha\beta} \\
&= (\delta_{i\alpha} \partial_\beta - \partial_i (\partial_k \partial_k)^\dagger \partial_\alpha \partial_\beta) \sigma_{\alpha\beta} \\
&= (\delta_{i\alpha} - \partial_i (\partial_k \partial_k)^\dagger \partial_\alpha) \partial_\beta \sigma_{\alpha\beta} \\
&= \pi_{i\alpha} \partial_\beta \sigma_{\alpha\beta},
\end{aligned} \tag{D.10}$$

which is the projected tensor-divergence of σ . \square

Note that we use the same symbol π for both the vector and stress tensor projectors. It should be clear from the context which version is used.

Appendix E. FVM of the incompressible Navier-Stokes equations in alternative notation

We here present the incompressible Navier-Stokes equations in an alternative notation to highlight how the RST in the FVM equation becomes non-local and non-symmetric. In Cartesian notation, the incompressible Navier-Stokes equations read

$$\partial_x u_x + \partial_y u_y + \partial_z u_z = 0, \tag{E.1}$$

$$\partial_t u_x + \partial_x (\sigma_{xx} + p) + \partial_y \sigma_{xy} + \partial_z \sigma_{xz} = 0, \tag{E.2}$$

$$\partial_t u_y + \partial_x \sigma_{yx} + \partial_y (\sigma_{yy} + p) + \partial_z \sigma_{yz} = 0, \tag{E.3}$$

$$\partial_t u_z + \partial_x \sigma_{zx} + \partial_y \sigma_{zy} + \partial_z (\sigma_{zz} + p) = 0, \tag{E.4}$$

where

$$\sigma := u \otimes u - v(\nabla u + \nabla u^T), \tag{E.5}$$

$$u \otimes u := \begin{pmatrix} u_x u_x & u_x u_y & u_x u_z \\ u_y u_x & u_y u_y & u_y u_z \\ u_z u_x & u_z u_y & u_z u_z \end{pmatrix}, \tag{E.6}$$

$$\nabla u := \begin{pmatrix} \partial_x u_x & \partial_y u_x & \partial_z u_x \\ \partial_x u_y & \partial_y u_y & \partial_z u_y \\ \partial_x u_z & \partial_y u_z & \partial_z u_z \end{pmatrix}. \tag{E.7}$$

In vector notation, we can write the equations as

$$\nabla \cdot u = 0, \quad \partial_t u + \nabla \cdot (\sigma + p\delta) = 0. \tag{E.8}$$

If we eliminate the pressure, this system can be written in projection-form as three equations

$$\partial_t u_x + \partial_x r_{xx} + \partial_y r_{xy} + \partial_z r_{xz} = 0, \quad (\text{E.9})$$

$$\partial_t u_y + \partial_x r_{yx} + \partial_y r_{yy} + \partial_z r_{yz} = 0, \quad (\text{E.10})$$

$$\partial_t u_z + \partial_x r_{zx} + \partial_y r_{zy} + \partial_z r_{zz} = 0, \quad (\text{E.11})$$

or, in vector notation,

$$\partial_t u + \nabla \cdot r = 0, \quad (\text{E.12})$$

where

$$r := \pi \sigma := \sigma + p \delta \quad (\text{E.13})$$

is the projected stress tensor,

$$p := -\Delta^\dagger \nabla \cdot \nabla \cdot \sigma, \quad (\text{E.14})$$

is a pressure field that depends non-locally on σ ,

$$\Delta := \partial_{xx} + \partial_{yy} + \partial_{zz}, \quad (\text{E.15})$$

is the Laplacian, $A^\dagger : b \mapsto x$ gives the unique solution (up to a constant) to the equation $Ax = b$ (we can for example design A^\dagger to return the solution with zero mean), and

$$\begin{aligned} \nabla \cdot \nabla \cdot \sigma := & \partial_{xx} \sigma_{xx} + \partial_{yy} \sigma_{yy} + \partial_{zz} \sigma_{zz} \\ & + 2(\partial_{xy} \sigma_{xy} + \partial_{xz} \sigma_{xz} + \partial_{yz} \sigma_{yz}). \end{aligned} \quad (\text{E.16})$$

E.1. The stress tensor in the FVM equation

Here we will write the FVM equation in “unresolved” form, without ever introducing the resolved part $\nabla^h \cdot r^h(\bar{u}^h)$ and the numerical stress tensor r^h . The goal is to understand the properties of the stress tensor.

The FVM equation is

$$\partial_t \bar{u}^h + \overline{\nabla \cdot \pi \sigma}^h = 0. \quad (\text{E.17})$$

For the tensor divergence, the discrete filter-swap property (92) can be written as $\overline{\nabla \cdot r}^h = \nabla^h \cdot \bar{r}^{h,*}$ for all tensors r , where $\nabla^h := (\partial_x^h, \partial_y^h, \partial_z^h)$ and

$$\bar{r}^{h,*} := \begin{pmatrix} \bar{r}_{xx}^{h,x} & \bar{r}_{xy}^{h,y} & \bar{r}_{xz}^{h,z} \\ \bar{r}_{yx}^{h,x} & \bar{r}_{yy}^{h,y} & \bar{r}_{yz}^{h,z} \\ \bar{r}_{zx}^{h,x} & \bar{r}_{zy}^{h,y} & \bar{r}_{zz}^{h,z} \end{pmatrix} \quad (\text{E.18})$$

contains the surface-averaging filters $f^{h,x} := g_x^h g_z^h$, $f^{h,y} := g_x^h g_z^h$, and $f^{h,z} := g_x^h g_y^h$ for 1D top-hat filters g_i^h in direction i . This gives the ∇^h -conservation law for \bar{u}^h :

$$\partial_t \bar{u}^h + \nabla^h \cdot \bar{\pi \sigma}^{h,*} = 0. \quad (\text{E.19})$$

The stress tensor appearing the ∇^h -conservation law for \bar{u}^h is therefore

$$\begin{aligned} \bar{\pi \sigma}^{h,*} &= \overline{\sigma + p \delta}^{h,*} \\ &= \begin{pmatrix} \bar{\sigma}_{xx}^{h,x} & \bar{\sigma}_{xy}^{h,y} & \bar{\sigma}_{xz}^{h,z} \\ \bar{\sigma}_{yx}^{h,x} & \bar{\sigma}_{yy}^{h,y} & \bar{\sigma}_{yz}^{h,z} \\ \bar{\sigma}_{zx}^{h,x} & \bar{\sigma}_{zy}^{h,y} & \bar{\sigma}_{zz}^{h,z} \end{pmatrix} + \begin{pmatrix} \bar{p}^{h,x} & 0 & 0 \\ 0 & \bar{p}^{h,y} & 0 \\ 0 & 0 & \bar{p}^{h,z} \end{pmatrix}. \end{aligned} \quad (\text{E.20})$$

This tensor is clearly non-symmetric, since

$$f^{h,x} \neq f^{h,y} \neq f^{h,z}. \quad (\text{E.21})$$

Furthermore, the surface-averaged pressure tensor

$$\bar{p \delta}^{h,*} = \begin{pmatrix} \bar{p}^{h,x} & 0 & 0 \\ 0 & \bar{p}^{h,y} & 0 \\ 0 & 0 & \bar{p}^{h,z} \end{pmatrix} \quad (\text{E.22})$$

is not an isotropic tensor (it cannot be written as $q^h \delta$ for some scalar field q^h). This is one of the reasons why the volume-averaged field \bar{u}^h is not ∇^h -divergence-free, i.e. $\nabla^h \cdot \bar{u}^h \neq 0$.

Define the ∇^h -divergence-free part of \bar{u}^h as

$$\bar{u}^{h,\pi} := \pi^h \bar{u}^h := \bar{u}^h - \nabla^h (\Delta^h)^\dagger \nabla^h \cdot \bar{u}^h, \quad (\text{E.23})$$

where π^h is a vector-version of a discrete pressure projector and $\Delta^h := \partial_{xx}^h + \partial_{yy}^h + \partial_{zz}^h$. The equation for $\bar{u}^{h,\pi}$ is

$$\partial_t \bar{u}^{h,\pi} + \nabla^h \cdot \pi^h \bar{\pi} \sigma^{h,*} = 0, \quad (\text{E.24})$$

where $\pi^h : \sigma \mapsto \sigma - (\Delta^h)^\dagger \nabla^h \cdot \sigma$ here denotes the tensor version of the discrete pressure projector (this should be clear from context). We used [Theorem 8](#) to swap ∇^h and π^h . In the tensor $\pi^h \bar{\pi} \sigma^{h,*}$, all the isotropic parts of $\bar{\pi} \sigma^{h,*}$ are absorbed by π^h . But since the tensor $\bar{\pi} \sigma^{h,*}$ is non-isotropic, it does not get absorbed, and cannot be removed from the expression $\pi^h \bar{\pi} \sigma^{h,*}$. This is also why the pressure projector π cannot be taken outside the surface-averaging filter. The stress tensor in the ∇^h -conservation law for $\bar{u}^{h,\pi}$ is therefore non-local in the external unresolved field u , which p depends on. This also remains the case even if we *reintroduce* the incompressibility constraint for $\bar{u}^{h,\pi}$ as

$$\nabla^h \cdot \bar{u}^{h,\pi} = 0, \quad \partial_t \bar{u}^{h,\pi} + \nabla^h \cdot \left(\bar{\pi} \sigma^{h,*} + q^h \delta \right) = 0, \quad (\text{E.25})$$

where q^h is the unique pressure (up to a constant) such that $\bar{\pi} \sigma^{h,*} + q^h \delta = \pi^h \bar{\pi} \sigma^{h,*}$ and thus $\bar{u}^{h,\pi}$ remains ∇^h -divergence-free. Importantly, $q^h \neq \bar{p}^h$ is not the filtered pressure field, since ∇^h -incompressibility and ∇ -incompressibility are different types of constraints. The original pressure field p is therefore still present in the stress tensor, even though the equations are in incompressibility-constraint-form.

The fact that the unresolved stress tensor in the equation for $\bar{u}^{h,\pi}$ is non-local in u also suggest that a closure model for this unresolved stress tensor should be non-local in $\bar{u}^{h,\pi}$. If $m(\bar{u}^{h,\pi}) \approx \bar{\pi} \sigma(u)^{h,*}$ is a closure model for the total unresolved stress tensor, then it should ideally be

1. non-symmetric ($m^T \neq m$),
2. non-local ($m(\bar{u}^{h,\pi})(x)$ should depend on $\bar{u}^{h,\pi}(x+d)$ for potentially large displacements $d \in \mathbb{R}^3$).

References

- [1] R. Verstappen, Merging Filtering, Modeling and Discretization to Simulate Large Eddies in Burgers' Turbulence, 2025.
- [2] L.C. Berselli, Mathematics of Large Eddy Simulation of Turbulent Flows. Scientific Computation, Berlin, Springer-Verlag, 2006.
- [3] S.B. Pope, Turbulent Flows, Cambridge; New York, Cambridge University Press, 2000.
- [4] P. Sagaut (Ed.), Large Eddy Simulation for Incompressible Flows: An Introduction. Scientific Computation, Berlin Heidelberg, Springer, third edition edition, 2006.
- [5] H. Lu, C.J. Rutland, Structural subgrid-scale modeling for large-eddy simulation: a review, Acta Mech. Sin. 32 (4) (2016) 567–578.
- [6] H.J. Bae, A. Lozano-Duran, Numerical and modeling error assessment of large-eddy simulation using direct-numerical-simulation-aided large-eddy simulation, 2022. arXiv:2208.02354
- [7] A. Beck, M. Kurz, Toward discretization-consistent closure schemes for large eddy simulation using reinforcement learning, Phys. Fluids 35 (12) (2023) 125122.
- [8] F.K. Chow, P. Moin, A further study of numerical errors in large-eddy simulations, J. Comput. Phys. 184 (2) (2003) 366–380.
- [9] S. Ghosal, An analysis of numerical errors in large-Eddy simulations of turbulence, J. Comput. Phys. 125 (1) (1996) 187–206.
- [10] J.W. Deardorff, On the magnitude of the subgrid scale eddy coefficient, J. Comput. Phys. 7 (1) (1971) 120–133.
- [11] B.J. Geurts, F.V.D. Bos, Numerically induced high-pass dynamics in large-eddy simulation, Phys. Fluids 17 (12) (2005) 125103.
- [12] X. Normand, M. Lesieur, Direct and large-eddy simulations of transition in the compressible boundary layer, Theor. Comput. Fluid Dyn. 3 (4) (1992) 231–252.
- [13] B. Vreman, B. Geurts, H. Kuerten, Comparison of numerical schemes in large-Eddy simulation of the temporal mixing layer, Int. J. Numer. Methods Fluids 22 (4) (1996) 297–311.
- [14] R.A. Clark, J.H. Ferziger, W.C. Reynolds, Evaluation of subgrid-scale models using an accurately simulated turbulent flow, J. Fluid Mech. 91 (1) (1979) 1–16.
- [15] T. Dairay, E. Lamballais, S. Laizet, J.C. Vassilicos, Numerical dissipation vs. subgrid-scale modelling for large eddy simulation, J. Comput. Phys. 337 (2017) 252–274.
- [16] S.D. Agdestein, B. Sanderse, Discretize first, filter next: learning divergence-consistent closure models for large-eddy simulation, J. Comput. Phys. 522 (2025) 113577.
- [17] J. Smagorinsky, General circulation experiments with the primitive equations: i. the basic experiment, Mon. Weather Rev. 91 (3) (1963) 99–164.
- [18] U. Schumann, Subgrid scale model for finite difference simulations of turbulent flows in plane channels and annuli, J. Comput. Phys. 18 (1975) 376–404.
- [19] T.S. Lund, The use of explicit filters in large eddy simulation, Comput. Math. Appl. 46 (4) (2003) 603–616.
- [20] R.S. Rogallo, P. Moin, Numerical simulation of turbulent flows, Annu. Rev. Fluid Mech. 16 (1984) 99–137.
- [21] M. Germano, U. Piomelli, P. Moin, W.H. Cabot, A dynamic subgrid-scale eddy viscosity model, Phys. Fluids A: Fluid Dyn. 3 (7) (1991) 1760–1765.
- [22] B. Vreman, B. Geurts, H. Kuerten, Large-eddy simulation of the turbulent mixing layer, J. Fluid Mech. 339 (1997) 357–390.
- [23] G.S. Winckelmans, A.A. Wray, O.V. Vasilyev, H. Jeanmart, Explicit-filtering large-eddy simulation using the tensor-diffusivity model supplemented by a dynamic smagorinsky term, Phys. Fluids 13 (5) (2001) 1385–1403.
- [24] F.M. Denaro, What does finite volume-based implicit filtering really resolve in large-Eddy simulations, J. Comput. Phys. 230 (10) (2011) 3849–3883.
- [25] F.M. Denaro, An inconsistency-free integral-based dynamic one- and two-parameter mixed model, J. Turbul. 19 (9) (2018) 754–797.
- [26] A. Jameson, The construction of discretely conservative finite volume schemes that also globally conserve energy or entropy, J. Sci. Comput. 34 (2) (2008) 152–187.
- [27] C. Fureby, F.F. Grinstein, Large eddy simulation of high-Reynolds-Number free and wall-Bounded flows, J. Comput. Phys. 181 (1) (2002) 68–97.
- [28] R. Maulik, O. San, Explicit and implicit LES closures for burgers turbulence, J. Comput. Appl. Math. 327 (2018) 12–40.
- [29] O. San, A.E. Staples, High-order methods for decaying two-dimensional homogeneous isotropic turbulence, Comput. Fluids 63 (2012) 105–127.
- [30] H.J. Bae, A. Lozano-Duran, Towards exact subgrid-scale models for explicitly filtered large-eddy simulation of wall-bounded flows, 2017.
- [31] D. Lilly, The representation of small-scale turbulence in numerical simulation experiments, Technical Report, Technical report, UCAR/NCAR, 1966.
- [32] F.H. Harlow, J.E. Welch, Numerical calculation of time-Dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (12) (1965) 2182–2189.
- [33] Y. Morinishi, T.S. Lund, O.V. Vasilyev, P. Moin, Fully conservative higher order finite difference schemes for incompressible flow, J. Comput. Phys. 143 (1) (1998) 90–124.
- [34] E. Hairer, M. Roche, C. Lubich, The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods, 1409, Springer, Berlin; Heidelberg, 1989.
- [35] B. Sanderse, Energy-conserving runge-Kutta methods for the incompressible navier-Stokes equations, J. Comput. Phys. 233 (2013) 100–131.
- [36] D. Kochkov, J.A. Smith, A. Alieva, Q. Wang, M.P. Brenner, S. Hoyer, Machine learning-accelerated computational fluid dynamics, Proc. Natl. Acad. Sci. 118 (21) (2021) 2101784118.
- [37] B. Sanderse, P. Stinis, R. Maulik, S.E. Ahmed, Scientific machine learning for closure models in multiscale problems: a review, Found. Data Sci. 7 (1) (2025) 298–337.

- [38] A.J. Chorin, Numerical solution of the navier-Stokes equations, *Math. Comput.* 22 (104) (1968) 745–762.
- [39] R. Témam, Sur l'approximation de la solution des équations de navier-Stokes par la méthode des pas fractionnaires (i), *Analysis* 32 (2) (1969) 135–153. *Archive for Rational Mechanics and*
- [40] A.W. Vreman, The projection method for the incompressible navier-Stokes equations: the pressure near a no-slip wall, *J. Comput. Phys.* 263 (2014) 353–374.