Journal of Computational Physics 499 (2024) 112721

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Fast macroscopic forcing method

Spencer H. Bryngelson [a,b,*,1], Florian Schäfer [a,1], Jessie Liu [c], Ali Mani [c]

[a] School of Computational Science & Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
[b] Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
[c] Department of Mechanical Engineering, Stanford University, Stanford, CA 94305, USA

ARTICLE INFO

Code available at: https://github.com/comp-physics/fast-mfm

Keywords:
Multi-scale modeling
Eddy diffusivity
Turbulence modeling
Operator recovery
Numerical homogenization

ABSTRACT

The macroscopic forcing method (MFM) of Mani and Park [1] and similar methods for obtaining turbulence closure operators, such as the Green's function-based approach of Hamba [2], recover reduced solution operators from repeated direct numerical simulations (DNS). MFM has already been used to successfully quantify Reynolds-averaged Navier–Stokes (RANS)-like operators for homogeneous isotropic turbulence and turbulent channel flows. Standard algorithms for MFM force each coarse-scale degree of freedom (i.e., degree of freedom in the RANS space) and conduct a corresponding fine-scale simulation (i.e., DNS), which is expensive. We combine this method with an approach recently proposed by Schäfer and Owhadi [3] to recover elliptic integral operators from a polylogarithmic number of matrix–vector products. The resulting Fast MFM introduced in this work applies sparse reconstruction to expose local features in the closure operator and reconstructs this coarse-grained differential operator in only a few matrix–vector products and correspondingly, a few MFM simulations. For flows with significant nonlocality, the algorithm first "peels" long-range effects with dense matrix–vector products to expose a more local operator. We demonstrate the algorithm's performance for the eddy diffusivity and eddy viscosity operators, which correspond to the unclosed parts of the ensemble-averaged transport equations, excluding the analytically known, closed parts of such equations. However, the algorithm can also be applied to the full operators. We focus on scalar transport in a laminar channel flow and momentum transport in a turbulent channel flow. For these problems, we recover eddy diffusivity- and eddy viscosity-like operators, respectively, at 1% of the cost of computing the exact operator via a brute-force approach for the laminar channel flow problem and 13% for the turbulent one. Applying these operators to compute the averaged fields of interest has visually indistinguishable behavior from the exact solution. Our results show that a similar number of simulations are required to reconstruct the operators to the same accuracy under grid refinement. Thus, the accuracy corresponds to the physics of the problem, not the numerics, so long as the grid is sufficiently refined. We glean that for problems in which the RANS space is reducible to one dimension, eddy diffusivity, and eddy viscosity operators can be reconstructed with reasonable accuracy using only a few simulations, regardless of simulation resolution or degrees of freedom.

* Corresponding author.
E-mail addresses: shb@gatech.edu (S.H. Bryngelson), florian.schaefer@cc.gatech.edu (F. Schäfer).
[1] These authors contributed equally to this work (alphabetical by last name).

https://doi.org/10.1016/j.jcp.2023.112721
Received 27 June 2023; Received in revised form 1 December 2023; Accepted 17 December 2023

Available online 21 December 2023
0021-9991/

1. Introduction

Significant computational resources are required to solve fluid dynamics problems; for example, [4] discuss resolution requirements for a turbulent boundary layer with application to a simulation of a ship hull [5]. Reduced-complexity surrogate models are a successful approach to reducing these costs. Historically, physical insight and analytical techniques have been used to develop these models, for example, Reynolds-averaged Navier–Stokes (RANS) closure models [6] for spatially or temporally averaged quantities. However, RANS models [7–9], often rely on ad hoc modeling assumptions [10] that are invalid for complex flows [11–13]. Data-driven approaches are emerging as semi-automated tools for developing reduced-complexity surrogate models. Some approaches attempt to represent the time evolution of the physical system via neural networks, a formidable task that involves reducing the entire Navier–Stokes operator [14,15]. An alternative approach is to compute effective operators that act on spatial or temporal averages, e.g., exact RANS closure operators. The governing equations are projected into the reduced or averaged space, and a forcing function is applied to examine the effect of the underlying fluctuations on the averaged behavior. Kraichnan [16] and Hamba [2] examined Green's function solutions (i.e., using Dirac-delta-function-type forcing) to scalar and momentum transport equations to develop exact expressions for closure operators. Similarly, the macroscopic forcing method (MFM) of Mani and Park [1] quantifies closure operators exactly by examining forcing and averaged responses input–output pairs. However, as a linear-algebra-based technique, MFM does not require the use of Dirac delta functions as forcing basis functions, and other functions, like polynomials [17] and harmonic functions [18], can be used.

MFM has been successfully applied to close reacting flow equations [19,20] and analyze homogeneous isotropic turbulence [18] and turbulent channel flow [21]. MFM is analogous to numerical homogenization, or the finite-dimensional approximation of solution spaces of partial differential equations (PDEs) [22]. These techniques amount to *operator recovery* or *learning*, where an unknown operator is estimated from a set of input–output pairs obtained from full-resolution simulations. These simulations are computationally expensive, so there is a pressing need to reduce the number of samples required, which we address in this work.

Using MFM, one constructs effective operators, or *macroscopic* operators, acting on solution averages from full-resolution simulations, called direct numerical simulations (DNS) of the governing or *microscopic* equations [1]. One can remove the known closed operators for RANS closure and equivalently formulate the macroscopic operators as eddy diffusivity (or eddy viscosity) operators. If the macroscopic operators are linear, the MFM procedure is no different from estimating a matrix from a limited number of matrix–vector products, where each matrix–vector product corresponds to a microscopic simulation (i.e., DNS). In a brute force MFM procedure, the number of microscopic simulations required to recover the macroscopic operator exactly equals the number of macroscopic degrees of freedom, which can be prohibitively large for many simulation problems, like high-Reynolds number turbulence.

One can partially address this problem by working in Fourier space [1] or fitting a parametric model to approximate the eddy diffusivity operator [21,23]. However, the former requires spatial homogeneity, and the latter's accuracy depends on the parametric model's quality. Liu et al. [17] introduces an improved model that uses the nonlocal eddy diffusivity operator's moments to approximate the operator. While these are viable approaches and the subject of ongoing work, this work aims to reconstruct the entire discretely defined nonlocal eddy diffusivity operator instead of prescribing or modeling its shape.

For many flows of practical interest, the nonlocal effects of closure terms show diffusive behavior. Thus, work on operator recovery for elliptic PDEs is closely related to MFM. [24] propose a "peeling" approach for recovering hierarchical matrices from a polylogarithmic number of matrix–vector products, although without rigorous bounds on the approximation error. Extensions of this algorithm were proposed by [25–27]. In this setting, eigendecompositions and randomized linear algebra have been used to recover elliptic solution operators from matrix–vector products [28–31]. Since the eigenvalues of elliptic solution operators follow a power law, these methods require $\mathrm{poly}(1/\varepsilon)$ matrix–vector products to obtain an ε approximation of the operator. In contrast, Schäfer et al. [32] showed that hidden sparsity of the solution operator results in an ε approximation from only $\mathrm{poly}(\log(1/\varepsilon))$ carefully crafted matrix–vector products. This speedup amounts to an *exponential reduction* in the number of matrix–vector products. The sparsity used by Schäfer and Owhadi [3] results from the locality of the partial differential operator shared by local fluid models.

We use this approach to accelerate the MFM to create the *Fast MFM*. The Fast MFM reveals the locality of the physical models to reduce the sample complexity of standard MFM operator recovery.

We apply Fast MFM to inhomogeneous and turbulent problems and reconstruct the RANS closure operators. Specifically, we consider passive scalar transport in a laminar 2D channel flow following Mani and Park [1] and reconstruct the corresponding eddy diffusivity operator, and momentum transport in a canonical turbulent 3D channel flow at $\mathrm{Re}_\tau = 180$ following Hamba [33] and Park and Mani [21] and reconstruct the corresponding eddy viscosity operator. These examples display sufficient spatio-temporal richness in their dynamics to argue that the Fast MFM can be applied more broadly. For example, one could tackle the open closure problems associated with multiphase flows [34–37]; indeed, multiphase flow has already been considered in this context for the Rayleigh–Taylor instability [38]. However, we do not show such examples in this work.

We briefly introduce MFM and similar approaches for recovering turbulence closure operators in section 2. Section 3 details the mathematical foundations of the sparse reconstruction procedure, and section 3.5 applies it to MFM with an extension to nonsymmetric operators, resulting in the Fast MFM. Results are presented in section 4, focusing on the 2D and 3D problems analyzed by Mani and Park [1] and Park and Mani [21]. Section 5 discusses the outlook of sparse reconstruction methods like the one presented for other flow problems and PDEs broadly.

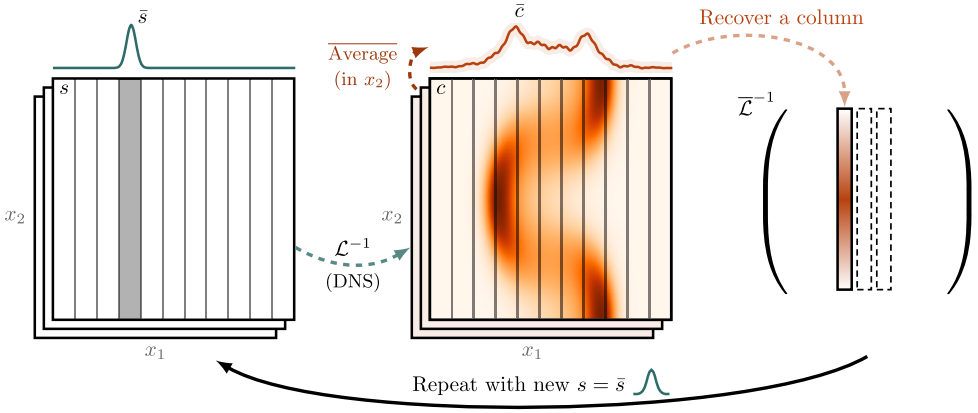

Fig. 1. Schematic of the MFM.

2. Background on the macroscopic forcing method (MFM)

2.1. The macroscopic forcing method

Given a set of linear microscopic equations,

$$\mathcal{L}c = 0, \tag{1}$$

and an averaging operator $c \mapsto \bar{c}$, the macroscopic (averaged) operator $\overline{\mathcal{L}}$ is defined to satisfy, for all microscopic solutions c of (1),

$$\overline{\mathcal{L}}\bar{c} = 0. \tag{2}$$

Often, (1) are advection-diffusion equations for scalar transport or linearizations of nonlinear PDEs such as Navier–Stokes equations, so the averaging operation can be written as

$$\bar{c}(x_1) = \frac{1}{L_2 \cdots L_{N_d}} \int_{\Omega_2} \cdots \int_{\Omega_{N_d}} c(x_1, \ldots x_{N_d})\, \mathrm{d}x_2 \cdots \mathrm{d}x_{N_d}, \tag{3}$$

where

$$\Omega = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_{N_d} \tag{4}$$

is the physical (possibly spatio-temporal) domain and L_i are the lengths in each coordinate direction x_i, $i \in \{1, \ldots, N_d\}$. In this example, the averaged (2) is a univariate problem in the non-averaged coordinate x_1 and is similar to the averaging in the steady laminar channel example problem in section 4.1. In the turbulent channel example problem in section 4.2, we average over temporal and homogeneous spatial directions. However, we point out that the techniques outlined in this work apply to a wider range of possible averaging operations, for example, closure operators for turbulent multiphase flow, but we leave such discussion for the conclusion.

Using MFM, one can determine the exact linear operator $\overline{\mathcal{L}}$ that acts on averages of flow statistics [1]. They infer this operator by adding a macroscopic forcing, s, to the right-hand sides of (1) and (2), with the property $s = \bar{s}$. Then by generating solution pairs to $\overline{\mathcal{L}}\bar{c} = \bar{s}$, obtained from solving the microscopic equations with forcing s and microscopic solution c (and corresponding $\bar{s}$ and averaged $\bar{c}$), they recover $\overline{\mathcal{L}}^{-1}$.

Fig. 1 shows an example MFM procedure schematically for a two-dimensional problem with coordinate directions x_1 and x_2. The relevant averaging direction is x_2, with averaged "strips" indicating averaging. This solution is forced by a field $s(x_1, x_2)$ as a Dirac delta function at a specific x_1 coordinate equivalent to its averaged field $\bar{s}(x_1)$. The inverse solution operator $\mathcal{L}^{-1}$ solves the problem (1) for $c(x_1, x_2)$ given $\bar{s}(x_1)$. This is computationally equivalent to solving the full-resolution system (1) or DNS. The averaged solution field $\bar{c}(x_1)$ corresponds to a column ("recover a column") of a macroscopic solution operator $\overline{\mathcal{L}}^{-1}$ under this averaging scheme. This procedure is repeated for all non-averaged degrees of freedom.

In this example, the non-averaged coordinate is x_1, so each discretized $x_{1,i}$ is locally forced (via a Dirac delta function) with $s = \bar{s}$. For example, discretely specifying $\bar{s} = [1\ 0 \ldots 0]^\top$ (a Dirac delta function at $x_{1,1}$), solving the microscopic equations for c, and averaging to obtain $\bar{c}$ leads to the first column of $\overline{\mathcal{L}}^{-1}$ via $\bar{c} = \overline{\mathcal{L}}^{-1}\bar{s}$. Discretely specifying $\bar{s} = [0\ 1\ 0 \ldots 0]^\top$ leads to the second column of $\overline{\mathcal{L}}^{-1}$, etc. Completing the MFM procedure gives access to the matrix representation $\overline{\mathcal{L}}$ via $s \mapsto \overline{\mathcal{L}}^{-1} s$. Since evaluating this map involves a high-resolution simulation, column-by-column construction of $\overline{\mathcal{L}}^{-1}$ is often intractable.

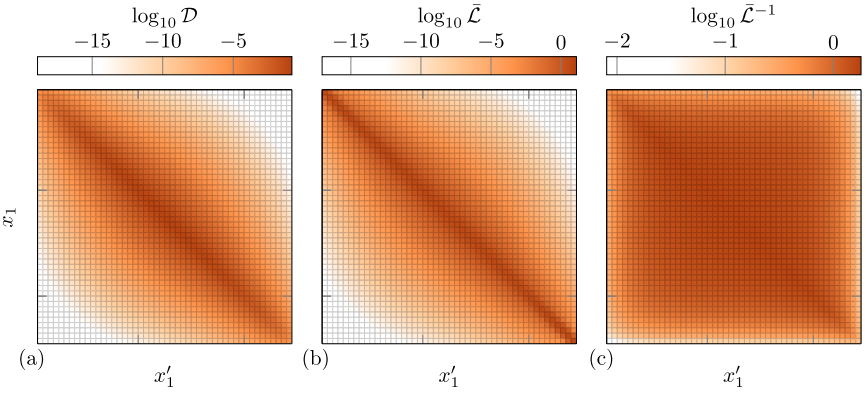

Fig. 2. Matrices D, $\overline{\mathcal{L}}$, and $\overline{\mathcal{L}}^{-1}$ as labeled for the 2D channel flow case with 50 non-averaged grid points (and so each is 50×50) for illustration purposes. The inverse operator matrix of (c) is nearly dense, though (a) and (b) are more strongly banded. Similar behavior is observed for finer discretizations, which correspond to larger matrices. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

2.2. The linear algebra of MFM

A linear algebraic perspective is useful for understanding MFM. To this end, we denote as $\mathcal{L}$ the matrix representation of a discretized advection-diffusion operator

$$\mathcal{L} = \frac{\partial}{\partial t} + u\cdot\nabla - \nabla\cdot(a\nabla), \tag{5}$$

where coefficients a, u, are allowed to vary in space and time. The inverse operator, $\mathcal{L}^{-1}$, takes a spatio-temporal forcing term s as input and returns the spatio-temporal field $c = \mathcal{L}^{-1}s$ by solving the PDE. Let $\boldsymbol{P}$ denote a projection onto coarse-scale features of interest, for example, spatio-temporal averages, and $\boldsymbol{E}$ denote an extension such that $\boldsymbol{P}\boldsymbol{E} = \mathcal{I}$, where $\mathcal{I}$ is the identity matrix. In the example of Fig. 1, rows of $\boldsymbol{P}$ correspond to averages of c in the x_2-direction of the domain, and the rows of $\boldsymbol{E}$ extend $\bar{s}$ to s. In other words, let $c(x_1, x_2)$ be discretized as a $N_1N_2\times1$ vector, $[c_{1,1}\ c_{2,1}\ldots c_{N_1,1}\ldots c_{N_1,N_2}]^\top$. Then $\boldsymbol{P}$ is a $N_1\times N_1N_2$ matrix that averages over the appropriate $c_{i,j}$, e.g., $\boldsymbol{P} = 1/N_2[\mathcal{I}\ \mathcal{I}\ldots\mathcal{I}]$ where $\mathcal{I}$ is a $N_1\times N_1$ identity matrix, and $\boldsymbol{E} = N_2\boldsymbol{P}^\top$. The macroscopic operator can then be expressed as [1]

$$\overline{\mathcal{L}} = \left(\boldsymbol{P}\mathcal{L}^{-1}\boldsymbol{E}\right)^{-1}, \tag{6}$$

where $\overline{\mathcal{L}}$ and $\mathcal{L}^{-1}$ are now discretized.

Another perspective on MFM can be obtained considering $\overline{\mathcal{L}}$ in discretized form. By using bases for its row and column space that consider the row and column spaces of $\boldsymbol{P}$ and $\boldsymbol{E}$, we obtain a 2×2 block matrix. After eliminating the second block, the macroscopic operator is obtained as the Schur complement of $\mathcal{L}$ [39, p. 19]:

$$\overline{\mathcal{L}} = \left(\left(\mathcal{L}^{-1}\right)_{1,1}\right)^{-1} = \mathcal{L}_{1,1} - \mathcal{L}_{1,2}\left(\mathcal{L}_{2,2}\right)^{-1}\mathcal{L}_{2,1}. \tag{7}$$

Computing $\mathcal{L}^{-1}$ or $\left(\mathcal{L}_{2,2}\right)^{-1}\mathcal{L}_{2,1}$ naïvely, column by column, requires as many solutions of the full-scale problem as there are coarse scale degrees of freedom.

2.3. Inverse MFM

As shown in Fig. 2, $\overline{\mathcal{L}}$ is more local than its computed inverse. Hoping to turn this locality into computational gains, Mani and Park [1] propose an inverse MFM to directly compute matrix–vector products with $\overline{\mathcal{L}}$ without first having to compute $\overline{\mathcal{L}}^{-1}$ using the ordinary MFM. This procedure can be interpreted as evaluating the right-hand side of (7) at the cost of solving a system of equations in $\mathcal{L}_{2,2}$. If (1) is an evolution PDE, this procedure can be interpreted as a control problem, where at each time step, the forcing s is chosen to maintain a target average $\bar{c}$. For example, at a given time step, n, the next c^{n+1} needs to satisfy $\bar{c}^{n+1} = \bar{c}$. One can time advance the governing equations without including the forcing and solve for an intermediate c^*, which may not have the requisite $\bar{c}$. The forcing is added in a correction step, $c^{n+1} = c^* + \bar{c} - \overline{c^*}$, such that $\bar{c}^{n+1}$ now satisfies $\bar{c}^{n+1} = \bar{c}$ and the implied s satisfies $s = \bar{s}$. For example, if a first-order explicit timestepping scheme is used, the implied s is $s = (\bar{c} - \overline{c^*})/\Delta t$, where Δt is the time step size.

2.4. The eddy diffusivity operator

In general $c \neq \bar{c}$, and the difference between the averaged field $\bar{c}$ and instantaneous field c (and analogously for the velocity field $\boldsymbol{u}$) can be expressed as

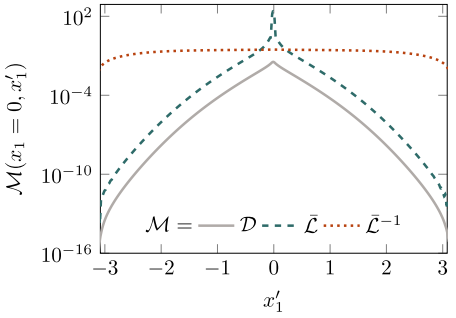

Fig. 3. The middle row of the matrices of Fig. 2 on a log scale. $\mathcal{D}$ and $\overline{\mathcal{L}}$ have a similar degree of locality, with entries decreasing in magnitude algebraically from the diagonal. The discretized operator $\overline{\mathcal{L}}^{-1}$ is dense; along its diagonal entries only decay modestly at the boundaries.

$$\boldsymbol{u} = \bar{\boldsymbol{u}} + \boldsymbol{u}' \quad \text{and} \quad c = \bar{c} + c', \tag{8}$$

where $(\cdot)'$ denotes fluctuations about the mean, and (8) is commonly-known as a Reynolds decomposition [40]. Substitution of (8) into $\mathcal{L}c = 0$ with the advection–diffusion operator in (5) and averaging results in the mean scalar equation:

$$\frac{\partial \bar{c}}{\partial t} + \bar{\boldsymbol{u}} \cdot \nabla \bar{c} + \nabla \cdot \overline{\boldsymbol{u}' c'} - \nabla \cdot (a \nabla \bar{c}) = 0. \tag{9}$$

The macroscopic operator, $\overline{\mathcal{L}}$, includes the closed temporal, mean advection, and mean molecular diffusion operators and a closure operator is needed for the scalar flux, $\overline{\boldsymbol{u}' c'}$. One may work with the closure operator, $\overline{\mathcal{L}}'$, where $\overline{\mathcal{L}}' \bar{c} = \nabla \cdot \overline{\boldsymbol{u}' c'}$ [1], or equivalently, the generalized eddy diffusivity [41]:

$$-\overline{u_i' c'}(\boldsymbol{x}, t) = \int_{\boldsymbol{x}', t'} \mathcal{D}_{ij}(\boldsymbol{x}, \boldsymbol{x}', t, t') \frac{\partial \bar{c}}{\partial x_j}\bigg|_{\boldsymbol{x}', t'} \mathrm{d}\boldsymbol{x}' \mathrm{d}t', \tag{10}$$

where $i, j \in \{1, \ldots, N_d\}$ are coordinate directions in the macroscopic space, and $\mathcal{D}_{ij}(\boldsymbol{x}, \boldsymbol{x}', t, t')$ is the eddy diffusivity kernel. Because the closed operators in (9) are known, recovering the generalized eddy diffusivity is equivalent to recovering the macroscopic operator, $\overline{\mathcal{L}}$.

For example, for the steady laminar channel problem in section 4.1, consider an averaging operation over all spatial directions except x_1. The steady mean scalar equation is:

$$\frac{\partial}{\partial x_1} \overline{u_1' c'} - a \frac{\partial^2 \bar{c}}{\partial x_1^2} = f, \tag{11}$$

where f is a source term in the laminar channel problem in (28), and the generalized eddy diffusivity is

$$-\overline{u_1' c'}(x_1) = \int_{x_1'} \mathcal{D}(x_1, x_1') \frac{\partial \bar{c}}{\partial x_1}\bigg|_{x_1'} \mathrm{d}x_1', \tag{12}$$

where we have simplified the notation, since $\mathcal{D} = \mathcal{D}_{11}$ is the only component that is active. Discretely,

$$-\overline{u_1' c'} = \mathcal{D} \frac{\partial \bar{c}}{\partial x_1}, \tag{13}$$

where $\overline{u_1' c'}$ is a $N_1 \times 1$ vector, $\mathcal{D}$ is a $N_1 \times N_1$ matrix, and $\partial \bar{c}/\partial x_1$ a $N_1 \times 1$ vector. The macroscopic operator, $\overline{\mathcal{L}}$, contains both the eddy diffusivity and the closed molecular diffusion operator:

$$\overline{\mathcal{L}} = -\frac{\partial}{\partial x_1} (\mathcal{D} + a\mathcal{I}) \frac{\partial}{\partial x_1}. \tag{14}$$

As shown in Fig. 2 and Fig. 3, the eddy diffusivity matrix is significantly more regular than $\overline{\mathcal{L}}$, making it a preferred target for an operator recovery strategy. Rather than using MFM with $\bar{c}$ and $\overline{\mathcal{L}}$, inverse MFM as described in section 2.3 can be used to specify $\partial \bar{c}/\partial x_1$ as discrete Dirac delta functions and recover $\mathcal{D}$. For example, specifying $\partial \bar{c}/\partial x_1 = [1\ 0 \ldots 0]^\top$ in (13) and post-processing $-\overline{u_1' c'}$ from a DNS leads to the first column of $\mathcal{D}$. Rather than repeating the procedure for each column of $\mathcal{D}$, the objective of the present work is to recover $\mathcal{D}$, and thus $\overline{\mathcal{L}}$, from as few simulations as possible, as accurately as possible, through better choices of $\partial \bar{c}/\partial x_1$.

Commonly used models for the scalar flux rely on the Boussinesq approximation [10], which assumes that the mixing by the fluctuations is isotropic and the mixing length is much smaller than the length associated with the mean scalar gradient. Under this approximation, the leading term of the Taylor-series expansion around $x_1' = x_1$ is

$$-\overline{u_1' c'}(x_1) = \int_{x_1'} \mathcal{D}(x_1, x_1') \frac{\partial \bar{c}}{\partial x_1}\bigg|_{x_1'} \mathrm{d}x_1' \approx \int_{x_1'} \mathcal{D}(x_1, x_1') \mathrm{d}x_1' \frac{\partial \bar{c}}{\partial x_1}\bigg|_{x_1}, \tag{15}$$

and results in an isotropic and local eddy diffusivity:

$$-\overline{u_1' c'}(x_1) = \mathcal{D}^{\text{Boussinesq}}(x_1) \frac{\partial \bar{c}}{\partial x_1}, \tag{16}$$

where

$$\mathcal{D}^{\text{Boussinesq}}(x_1) = \int_{x_1'} \mathcal{D}(x_1, x_1') \mathrm{d}x_1'. \tag{17}$$

Discretely, this corresponds to only the diagonal of $\mathcal{D}$ being active. However, for many flows, the Boussinesq approximation is invalid [42]. As shown in Fig. 2, $\mathcal{D}$ is a full matrix, which motivates efficient recovery of $\mathcal{D}$ rather than using the Boussinesq approximation.

Thus, we will perform Fast (and brute force) MFM on $\mathcal{D}$, corresponding to an eddy diffusivity, and not $\overline{\mathcal{L}}$ because it has a more local matrix structure. In the case of our turbulence closures, $\overline{\mathcal{L}}$ has an additional diffusion that is already known and thus does not need to be *recovered*. So, we can recover $\mathcal{D}$ and then compute $\overline{\mathcal{L}}$ from $\mathcal{D}$ as needed. In cases with no such $\mathcal{D}$; for example, outside of the turbulence context, one can perform the same algorithm on the more general $\overline{\mathcal{L}}$ matrix.

2.5. The eddy viscosity operator

For generalization to momentum transport and the turbulent channel flow considered in section 4.2, the incompressible Navier–Stokes equations are

$$\frac{\partial u_j}{\partial t} + \frac{\partial u_i u_j}{\partial x_i} = -\frac{\partial p}{\partial x_j} + \frac{1}{\mathrm{Re}} \frac{\partial^2 u_j}{\partial x_i \partial x_i} + r_j, \tag{18}$$

$$\frac{\partial u_j}{\partial x_j} = 0, \tag{19}$$

where Re is the Reynolds number, $\boldsymbol{r}$ is a body force, p is the pressure, and $\boldsymbol{u}$ are velocities. Application of the Reynolds decomposition in (8) and averaging leads to the RANS equations:

$$\frac{\partial \bar{u}_j}{\partial t} + \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_i} = -\frac{\partial \bar{p}}{\partial x_j} + \frac{1}{\mathrm{Re}} \frac{\partial^2 \bar{u}_j}{\partial x_i \partial x_i} + \frac{\partial}{\partial x_i} \overline{u_i' u_j'} + \bar{r}_j, \tag{20}$$

$$\frac{\partial \bar{u}_j}{\partial x_j} = 0, \tag{21}$$

in which a closure operator is needed for the Reynolds stress tensor, $\overline{u_i' u_j'}$.

As with passive scalar transport in section 2.4, widely-used RANS models [7–9] rely on the Boussinesq approximation [10]:

$$-\overline{u_i' u_j'} = \nu_{\mathrm{T}} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) - \frac{2}{3} k \delta_{ij}, \tag{22}$$

where ν_{T} is a local and isotropic eddy viscosity, $2k \equiv \overline{u_i' u_i'}$ is the turbulent kinetic energy, and δ_{ij} is the Kronecker delta. The turbulent kinetic energy is added such that the trace of (22) is consistent but is often omitted from models where it is not readily available. The Boussinesq approximation is known to be inadequate for complex flows [11–13]. The generalized eddy viscosity is given by [33]:

$$-\overline{u_i' u_j'}(\boldsymbol{x}, t) = \int_{\boldsymbol{x}', t'} \mathcal{D}_{ijkl}(\boldsymbol{x}, \boldsymbol{x}', t, t') \frac{\partial \bar{u}_l}{\partial x_k}\bigg|_{\boldsymbol{x}', t'} \mathrm{d}\boldsymbol{x}' \mathrm{d}t', \tag{23}$$

where $i, j, k, l \in \{1, \ldots, N_d\}$ are the coordinate directions in the macroscopic space.

Because there are not enough degrees of freedom to use inverse MFM to simultaneously specify the mean velocity gradient and post-process the Reynolds stresses, following Mani and Park [1], the generalized momentum transport equations are used for MFM:

$$\frac{\partial v_j}{\partial t} + \frac{\partial u_i v_j}{\partial x_i} = -\frac{\partial q}{\partial x_j} + \frac{1}{\mathrm{Re}} \frac{\partial^2 v_j}{\partial x_i \partial x_i} + s_j, \tag{24}$$

$$\frac{\partial v_j}{\partial x_j} = 0, \tag{25}$$

where the velocity u_i is computed from the incompressible Navier–Stokes equations and q is a pressure-like quantity that ensures incompressibility of the vector field v_j. In this formulation, the generalized eddy viscosity is

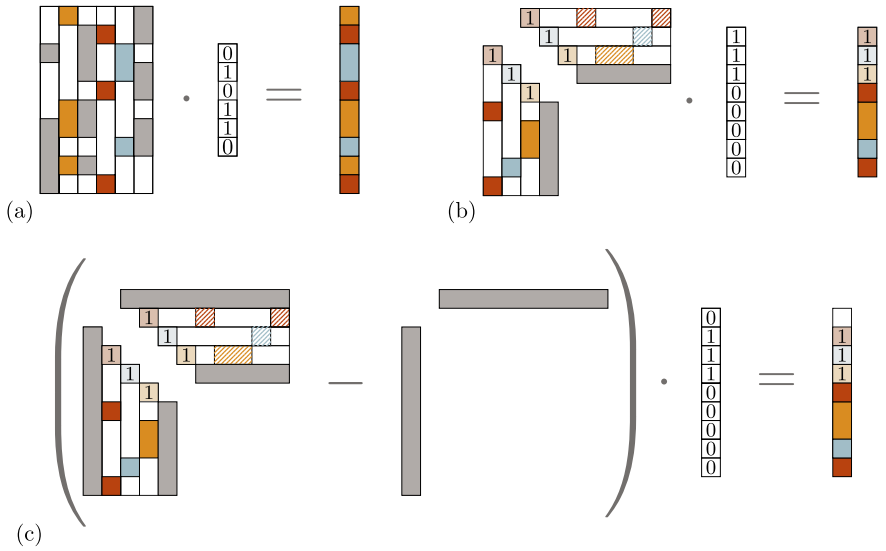

Fig. 4. (a) Sparse recovery: Columns with non-overlapping but known sparsity patterns (shown in color) can be recovered by a single matrix–vector product via a carefully chosen vector. (b) Factorization: Cholesky factorizations with leading-column sparsity patterns can be recovered similarly. (c) Peeling: If denser columns of the factorization prevent recovery of sparser ones, identify dense columns first and subtract their contribution to recover the sparser ones. © 2023 Schäfer and Owhadi.
This figure was adapted from Schäfer and Owhadi [3] with author permission.

$$-\overline{u_i' v_j'}(\boldsymbol{x},t) = \int_{\boldsymbol{x}',t'} \mathcal{D}_{ijkl}(\boldsymbol{x},\boldsymbol{x}',t,t') \frac{\partial \bar{v}_l}{\partial x_k}\bigg|_{\boldsymbol{x}',t'} \mathrm{d}\boldsymbol{x}'\mathrm{d}t' \tag{26}$$

and is equivalent to the generalized eddy viscosity in (23) [21,33].

Similar to computing the eddy diffusivity in section 2.4, a brute force approach to recover the eddy viscosity is to specify various components of $\partial \bar{v}_l/\partial x_k$ as Dirac delta functions using inverse MFM and post-process $-\overline{u_i' v_j'}$. When used in this manner, MFM is equivalent to the Green's function-based approach of Hamba [33] as discussed by Liu et al. [17]. Efficient recovery of the discretized eddy viscosity using the Fast MFM reduces the number of matrix–vector products, and thus the number of DNSs, by choosing carefully crafted $\partial \bar{v}_l/\partial x_k$ rather than using brute force Dirac delta functions.

3. LU reconstruction of elliptic operators

3.1. Reconstructing elliptic operators from matrix–vector products

We use the LU variant of the *Cholesky reconstruction* of Schäfer and Owhadi [3] to construct the eddy diffusivity operator. Schäfer and Owhadi [3] prove that the solution operators of divergence form elliptic partial differential equations in dimension N_d can be reconstructed to accuracy ϵ from only $\mathcal{O}\left(\log^{N_d+1}(\epsilon^{-1})\right)$ solutions for carefully selected forcing terms. We briefly review this approach, which forms the basis of this work.

3.2. Graph coloring

Graph coloring allows one to reconstruct multiple columns of a sparse matrix from a single matrix–vector product. The key idea is to identify groups of columns with non-overlapping sparsity sets and use a right-hand side that only activates those columns. As illustrated in Fig. 4, the selected columns can be read off from the resulting matrix–vector product. Similarly, graph coloring can also reveal the leading columns of a sparse LU factorization. Once a row-column pair of the LU factors are identified, it can be used to correct the matrix–vector products to reveal later columns. This procedure, a variant of which was first proposed by Lin et al. [24], is referred to as peeling (Fig. 4).

3.3. Cholesky factors in wavelet basis

It is well-known that the solution operators of elliptic PDEs are dense, owing to the long-range interactions produced by diffusion. However, Schäfer et al. [32] show that when represented in a multiresolution basis ordered *from coarse to fine*, solution operators of elliptic PDEs have almost sparse Cholesky factors. This phenomenon is illustrated in Fig. 5. The leading columns of the Cholesky factors, corresponding to a coarse-scale basis function with global support, are dense and, therefore, limit the efficiency of graph coloring. However, they are few and can be identified efficiently and removed via peeling. This procedure can be repeated to reveal

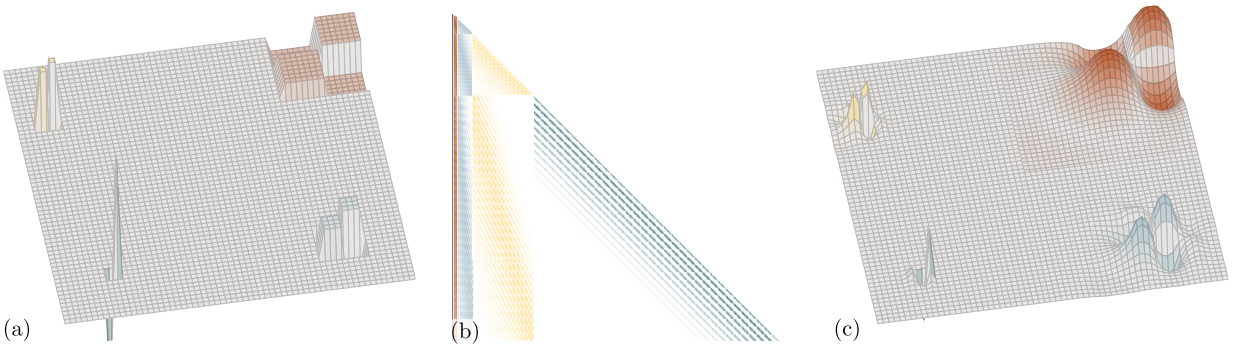

Fig. 5. (a) shows basis function on four different scales of a multiresolution basis. (b) shows the decay pattern of the Cholesky factorization of an elliptic Green's function discretized in this basis. (c) shows four columns of the Cholesky factor as spatial functions. © 2023 Schäfer and Owhadi.
This figure was reproduced from Schäfer and Owhadi [3] with author permission.

progressively finer columns. The growing number of basis functions on finer scales is compensated for by their smaller support and, thus, increased gains due to graph coloring. Thus, the number of matrix–vector products required is approximately constant across levels. The resulting procedure is described in Algorithm 1. As described in Schäfer and Owhadi [3], the operation $\texttt{scatter}_c$ takes in the vector obtained from a peeled matrix–vector product computed in lines 5 and 6 and uses it to recover the columns associated with the color c. In principle, further compression of the resulting operator is possible using the techniques of Schäfer et al. [43].

Algorithm 1 LU reconstruction in wavelet basis given by $\mathbf{W}$ with measurements given by $\mathbf{M}$.

1: $\mathbf{L} \leftarrow 0 \times N$ empty matrix
2: $\mathbf{U} \leftarrow N \times 0$ empty matrix
3: $\mathbf{D} \leftarrow 0 \times 0$ empty diagonal matrix
4: **for** c color **do**
5: $\quad \mathbf{L}_{\text{new}} \leftarrow \texttt{scatter}_c\left(\mathbf{W}^\top D\mathbf{M}_{:,c} - \mathbf{L}\mathbf{D}\mathbf{U}\mathbf{W}^\top\mathbf{M}_{:,c}\right)$
6: $\quad \mathbf{U}_{\text{new}} \leftarrow \texttt{scatter}_c\left(\mathbf{W}^\top D\mathbf{M}_{:,c} - \mathbf{U}^\top\mathbf{D}\mathbf{L}^\top\mathbf{W}^\top\mathbf{M}_{:,c}\right)$
7: $\quad \mathbf{D}_{\text{new}} \leftarrow \operatorname{diag}\left(\left(\mathbf{L}_{\text{new}} + \mathbf{U}_{\text{new}}\right)/2\right)^{-1}$
8: $\quad \mathbf{L}, \mathbf{U}, \mathbf{D} \leftarrow \texttt{hcat}\left(\mathbf{L}, \mathbf{L}_{\text{new}}\right), \texttt{vcat}\left(\mathbf{U}, \mathbf{U}_{\text{new}}\right), \texttt{dcat}\left(\mathbf{D}, \mathbf{D}_{\text{new}}\right)$ {Concatenate to prior solution}
9: **end for**
10: $\mathbf{L} \leftarrow \mathbf{L}\mathbf{D}$
11: **return** approximate LU factorization $\mathbf{L}\mathbf{U}$ of $\mathbf{W}^\top D\mathbf{W}$

3.4. Adaptation to fast MFM

The eddy diffusivity operator is not a divergence-form elliptic solution operator. In particular, it is not symmetric. Instead of the Cholesky recovery of Schäfer and Owhadi [3], we use an LU recovery that recovers a sparse LU factorization of the target matrix. Columns of L are recovered from matrix–vector products, and rows of U are recovered from matrix-transpose–vector products (transpose–vector products), which can be computed by solving the adjoint equation of $\mathcal{L}$. No rigorous guarantees exist for the accuracy of LU reconstruction applied to eddy diffusivity matrices. However, Schäfer et al. [32] show a wide range of diffusion-like operators, including those produced by fractional-order Matérn or Cauchy kernels, produce sparse Cholesky factors, despite the lack of theory supporting this observation.

3.5. Fast MFM on nonsymmetric operators

As remarked by Schäfer and Owhadi [3], a LU version of Cholesky reconstruction that extends to nonsymmetric matrices requires matrix–vector products and transpose–vector products. Similar requirements arise in hierarchical low-rank approaches [24,44]. Schur complementation commutes with transposition in the sense that

$$\left(\overline{\mathcal{L}}\right)^\top = \left(\left(\mathcal{L}^{-\top}\right)_{1,1}\right)^{-1} = \left(\mathcal{L}^\top\right)_{1,1} - \left(\mathcal{L}^\top\right)_{1,2}\left(\left(\mathcal{L}^\top\right)_{2,2}\right)^{-1}\left(\mathcal{L}^\top\right)_{2,1}. \tag{27}$$

As a result, transpose–vector products with $\overline{\mathcal{L}}$ can be obtained by applying (inverse or forward) MFM to the $\overline{\mathcal{L}}^\top$. In the case of the discretized advection-diffusion operator in (5), when the system is solved up to time T, the transpose of $\overline{\mathcal{L}}$ can be obtained by replacing $v(x,t)$ with $-v(x,T-t)$, $a(x,t)$ with $a(x,T-t)$, and by using the solution at time T as the initial condition. Here x is the spatial coordinate, and t denotes time. The resulting PDE is often called the adjoint problem and frequently arises in the computation of sensitivities of solutions of PDEs with respect to their coefficients, boundary, and initial conditions. We empirically validate our method using matrix–vector products and transpose–vector products obtained from a full eddy diffusivity operator constructed via brute force (column-by-column) MFM. We leave an adjoint-based MFM that efficiently implements transpose–vector products as future work.

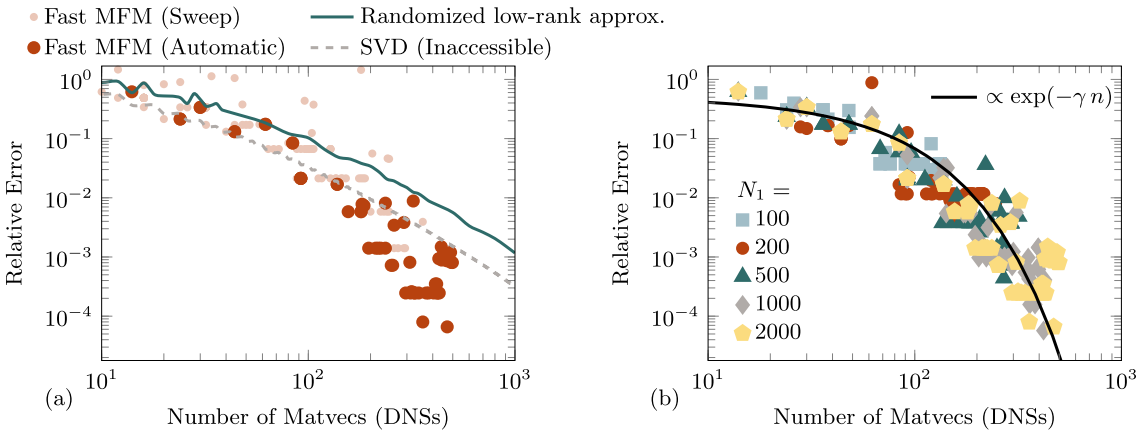

Fig. 6. Relative L_2 errors between the exact MFM operator D and its reconstruction for the laminar flow configuration. In (b), only the Fast MFM with automatically chosen parameterization is shown, but for different resolutions N_1 and the number of matrix–vector products is denoted by n in the legend.

4. Results

4.1. Steady-state laminar channel flow

Consider a 2D domain representing a channel with left and right walls at $x_1 = \pm\pi$ with Dirichlet boundary condition $c = 0$ and the top and bottom walls at $x_2 = 0, 2\pi$ with no flux condition $\partial c/\partial x_2 = 0$. The scalar field $c(x_1, x_2)$ is governed by a steady advection–diffusion equation with a uniform source term

$$u_1 \frac{\partial c}{\partial x_1} + u_2 \frac{\partial c}{\partial x_2} = 0.05 \frac{\partial^2 c}{\partial x_1^2} + \frac{\partial^2 c}{\partial x_2^2} + 1, \tag{28}$$

where the unequal diffusion constants in the coordinate directions are an outcome of directional nondimensionalization. The flow is incompressible and satisfies no-penetration boundary conditions on the walls. The steady velocity field is prescribed as

$$u_1 = (1 + \cos(2x_1))\cos(2x_2), \quad u_2 = \sin(2x_1)\sin(2x_2). \tag{29}$$

The PDE is discretized on a uniform staggered mesh with N_1 and $N_2 = N_1/2$ grid points in the x_1 and x_2 coordinate directions. Second-order accurate central differences are used. The advective fluxes at the cell faces are computed via second-order interpolation and then multiplication of the divergence-free velocity at the face centers. At the cell faces $x_1 = \pm\pi$, the fluxes are computed using ghost points that enforce the specified Dirichlet boundary conditions, while at the cell faces at the top and bottom boundaries, the no flux condition is naturally enforced.

Fig. 6 shows the eddy diffusivity operator errors for the laminar flow configuration. In (a), the mesh resolution in the x_1 (non-averaged) coordinate is $N_1 = 2000$. Fast MFM errors are smaller than a truncated SVD reconstruction of the same operator. The latter provides the optimal low-rank approximation of D but requires access to the full operator, which is impractical. A randomized low-rank representation, which is available, is also shown. The errors for this reconstruction are about 10 times larger than the SVD. Compared to Fast MFM, these errors are also larger as the number of matrix–vector products increases. The Fast MFM requires choosing the distance between basis functions of the same color (see section 3.2) and the level at which the wavelet coefficients are truncated. The former parameter dictates the cost-accuracy trade-off, and choosing a suitable truncation can improve the method's cost and stability. A sweep over a wide range of parameters is shown in shaded markers. We use a heuristic to set these parameters, which results in the non-shaded darker marks. Sometimes, the heuristic produces poor parameter choices, resulting in larger Fast MFM errors.

In Fig. 6 (b), we perform a similar analysis but only show the Fast MFM results for varying mesh sizes N_1. Errors decrease exponentially with the number of matrix–vector products (corresponding to the number of DNSs) with about the same fit coefficients regardless of N_1. This indicates that, so long as the forward simulation is fully resolved and mesh-independent, the Fast MFM reconstruction is dependent on the *physical locality* of the operator, not a numerical or discretized one. Thus, operator recovery for high-resolution simulations has an out-sized benefit over traditional MFM.

Fig. 7 shows the application of the recovered operator of Fig. 6 to compute $\bar{c}$ using (14). The exact result is computed using $N_1 = 2000$ DNSs to recover $\bar{c}$. For fewer DNSs, 26 out of the 2000 total non-averaged degrees of freedom, the Fast MFM result matches the exact result well, but the other methods do not.

4.2. Turbulent channel flow

We next consider a fully-developed turbulent channel flow, reconstructing eddy viscosities as discussed in section 2.5. We consider

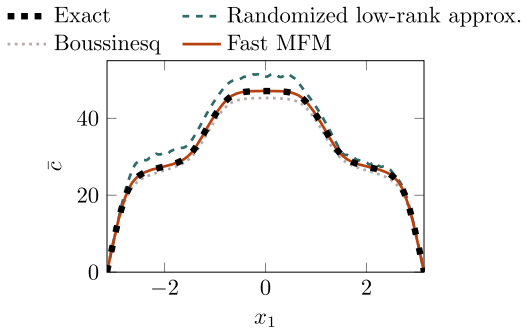

Fig. 7. Reconstruction of $\bar{c}$ for the laminar, steady channel flow problem. Results are shown for a randomized low-rank approximation, the Boussinesq approximation, and the Fast MFM result. This visualization is 26 out of 2000 possible samples.

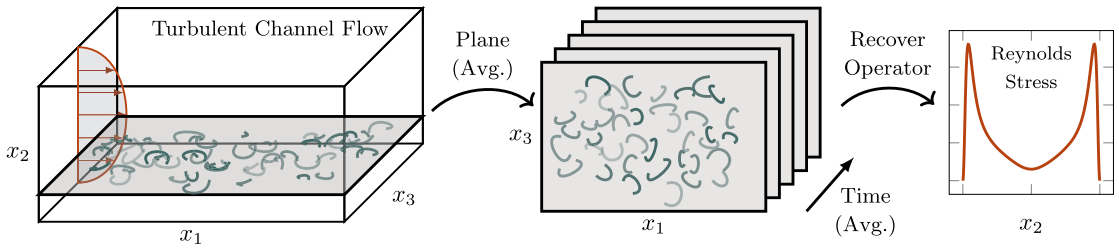

Fig. 8. Diagram of the Fast MFM reconstruction procedure for the 3D turbulent channel flow case.

a case with $\mathrm{Re}_\tau = u_\tau \delta/\nu = 180$ where δ is the channel half-width and u_τ is the friction velocity. The mean flow is in the x_1 direction, the x_2 direction is wall-normal, and the x_3 direction is the span with periodic boundaries. The streamwise domain length is 2π, and the spanwise length is π. The body force, $\boldsymbol{r}$, is the mean pressure gradient in the periodic simulation and is $\boldsymbol{r} = (1, 0, 0)$ in this nondimensionalized problem. To ensure statistical convergence, the incompressible Navier–Stokes equations are solved using direct numerical simulation with a 144^3 grid for $T = 500\delta/u_\tau$. Simulation-result baselines, including the discretized $\mathcal{D}$ and $\overline{\mathcal{L}}$ matrices, for this case, follow from Park and Mani [21] and are used herein. Fig. 8 shows our MFM procedure, averaging all independent variables except for the wall-normal coordinate x_2. We thus recover the Reynolds stresses as a function of x_2. Given $\partial \bar{v}_1/\partial x_2$ is the only active component of the velocity gradient tensor, the generalized eddy viscosity for this problem is

$$-\overline{u_i' v_j'}(x_2) = \int_{x_2'} \mathcal{D}_{ij21}(x_2, x_2') \frac{\partial \bar{v}_1}{\partial x_2}\bigg|_{x_2'} \mathrm{d}x_2'. \tag{30}$$

Discretely, for a given direction, e.g., $i = 2$ and $j = 1$:

$$-\overline{u_2' v_1'} = \mathcal{D}_{2121} \frac{\partial \bar{v}_1}{\partial x_2}, \tag{31}$$

where $-\overline{u_2' v_1'}$ is a $N_2 \times 1$ vector, $\mathcal{D}_{2121}$ is a $N_2 \times N_2$ eddy viscosity operator, and $\partial \bar{v}_1/\partial x_2$ is a $N_2 \times 1$ vector, where $N_2 = 144$.

For comparison, the anisotropic Boussinesq eddy viscosity corresponding to the leading term of the Taylor series expansion of (30) around $x_2' = x_2$ is

$$-\overline{u_i' v_j'}(x_2) = \mathcal{D}_{ij21}^{\text{Boussinesq}}(x_2) \frac{\partial \bar{v}_1}{\partial x_2}, \tag{32}$$

where

$$\mathcal{D}_{ij21}^{\text{Boussinesq}}(x_2) = \int_{x_2'} \mathcal{D}_{ij21}(x_2, x_2') \mathrm{d}x_2'. \tag{33}$$

Fig. 9 shows the errors in the recovered eddy viscosities. The errors are computed as the difference in operator norms between the approximate and exact solutions to the discretized problem. The exact solution is recovered via brute force inverse MFM, which independently computes each non-averaged degree of freedom to recover all columns of $\mathcal{D}_{ij21}$ for each direction i and j. In Fig. 9, the trends of (a) and (b) are similar, with the Fast MFM having smaller errors than both the SVD, which is inaccessible in practice, and a randomized low-rank approximation of it that is accessible. The differences in errors are small for small numbers of matrix–vector produces as the peeling procedure removes the long-range behaviors. For larger numbers of matrix–vector products, the difference

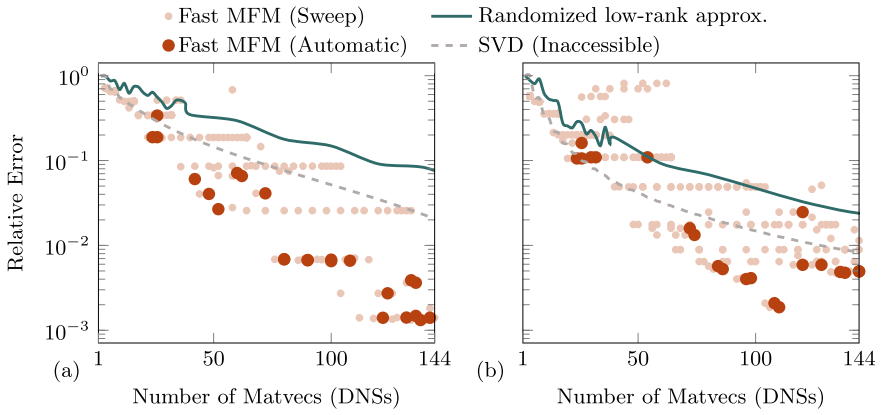

Fig. 9. Relative errors in the recovered eddy viscosity kernels (a) D_{2121} and (b) D_{1121} are shown for the $\mathrm{Re}_\tau = 180$ turbulent channel flow configuration.

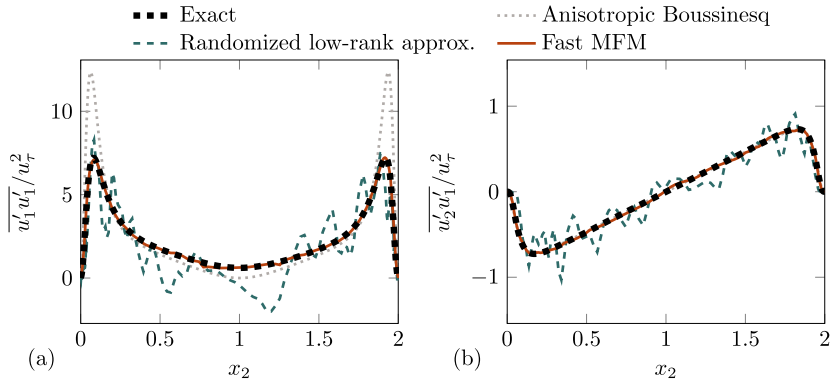

Fig. 10. Reynolds stress reconstructions (a) $\overline{u_1'u_1'}$ and (b) $\overline{u_2'u_1'}$ for the turbulent channel flow configuration. Results are shown for 20 out of 144 possible samples.

increases. Fast MFM has a factor of about 100 smaller errors than the low-rank approximation for 100 matrix–vector products in both (a) and (b).

Fig. 10 shows the Reynolds stress reconstructions for the turbulent channel flow. While direct computation of Reynolds stresses needs only one DNS, we use Reynolds stresses as a metric to assess the accuracy of the recovered eddy viscosity operator, which inevitably requires multiple simulations. The Fast MFM results (solid, thin line) are compared with the anisotropic Boussinesq approximation in (32) and a randomized low-rank procedure. Exact results are recovered via brute-force MFM. For 20 simulations, the difference between the exact solution and Fast MFM is not discernible. For the same number of simulations, the low-rank procedure does not produce a reasonable approximation for either component. The anisotropic Boussinesq approximation is a good one for the transverse stress component $\overline{u_2'u_1'}$ of Fig. 10 (b) but does poorly with the $\overline{u_1'u_1'}$ reconstruction in Fig. 10 (a).

5. Conclusion

This work uses a linear algebra approach to reconstruct closure operators. Fast MFM uses sparse recovery and peeling techniques, revealing local behaviors that crafted forcings can simultaneously recover. Results show that tens of simulations are required to reconstruct the eddy diffusivity operator and averaged field to visual accuracy. This contrasts against brute-force MFM, which forces each degree of freedom and is thus prohibitively expensive; the Boussinesq approximation, which is shown to have inaccuracies in some test cases; randomized low-rank approximations, which, while feasible, have poor accuracy; and even SVD, which performs worse than Fast MFM and is inaccessible in a simulation environment.

Other ongoing work focuses on modeling the nonlocal eddy diffusivity using partial differential equations and limited information about the exact eddy diffusivity. We showed that the Fast MFM procedure recovers full nonlocal eddy diffusivity operators at low sample complexity. It is a stepping stone toward the long-term goal of sample-efficient recovery of coarse-grained time integrators.

The Fast MFM formulation is natural for other turbulent flows. For example, higher Reynolds numbers can be analyzed in the same way. In such cases, the relative advantage of Fast MFM over traditional MFM will depend on the degree of locality in the flow. Multiphase flows can also be studied under the same strategy. One such previous example includes MFM analysis of the Rayleigh–Taylor instability [38]. Flows in complex geometries are also tractable, though they require particular attention to the boundary condition formulations.

CRediT authorship contribution statement

Spencer H. Bryngelson: Conceptualization, Investigation, Methodology, Project administration, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Florian Schäfer:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Writing – original draft, Writing – review & editing. **Jessie Liu:** Conceptualization, Formal analysis, Resources, Software, Writing – original draft, Writing – review & editing. **Ali Mani:** Conceptualization, Resources, Writing – original draft, Writing – review & editing.

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.

Data availability

The code used for this paper is available at: https://github.com/comp-physics/fast-mfm

Acknowledgements

This work used Bridges2 at the Pittsburgh Supercomputing Center through allocation TG-PHY210084 (PI Spencer Bryngelson) from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296. SHB also acknowledges the resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. SHB acknowledges support from the Office of the Naval Research under grant N00014-22-1-2519 (PM Dr. Julie Young). FS gratefully acknowledges support from the Office of Naval Research under grant N00014-23-1-2545 (PM Dr. Reza Malek-Madani). JL was supported by the Boeing Company. AM was supported by the Office of Naval Research under grant N00013-20-1-2718. The authors gratefully acknowledge Danah Park for providing the DNS and MFM data of the turbulent channel flow simulations and Dana Lavacot for fruitful discussions.

References

[1] A. Mani, D. Park, Macroscopic forcing method: a tool for turbulence modeling and analysis of closures, Phys. Rev. Fluids 6 (2021) 054607.
[2] F. Hamba, An analysis of nonlocal scalar transport in the convective boundary layer using the Green's function, J. Atmos. Sci. 52 (1995) 1084–1095.
[3] F. Schäfer, H. Owhadi, Sparse recovery of elliptic solvers from matrix–vector products, arXiv:2110.05351, 2023.
[4] X.I. Yang, K.P. Griffin, Grid-point and time-step requirements for direct numerical simulation and large-eddy simulation, Phys. Fluids 33 (2021) 015108.
[5] M. Liefvendahl, C. Fureby, Grid requirements for les of ship hydrodynamics in model and full scale, Ocean Eng. 143 (2017) 259–268.
[6] H. Tennekes, J.L. Lumley, A First Course in Turbulence, MIT Press, 1972.
[7] P. Spalart, S. Allmaras, A one-equation turbulence model for aerodynamic flows, in: 30th Aerospace Sciences Meeting and Exhibit, 1992, p. 439.
[8] F.R. Menter, Two-equation eddy-viscosity turbulence models for engineering applications, AIAA J. 32 (1994) 1598–1605.
[9] D.C. Wilcox, Turbulence Modeling for CFD, vol. 2, DCW Industries, La Canada, CA, 1998.
[10] J. Boussinesq, Essai sur la théorie des eaux courantes, in: Mémoires Présentés Par Divers Savants a l'Academie des Sciences de l'Institute National de France, vol. XXIII, Impr. Nationale, 1877.
[11] D.C. Jespersen, T.H. Pulliam, M.L. Childs, OVERFLOW Turbulence modeling resource validation results, Technical Report ARC-E-DAA-TN35216, 2016.
[12] A. Probst, R. Radespiel, C. Wolf, T. Knopp, D. Schwamborn, A comparison of detached-eddy simulation and Reynolds-stress modeling applied to the flow over a backward-facing step and an airfoil at stall, in: 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2010, p. 920.
[13] D. Park, J. Liu, A. Mani, Direct measurement of the eddy viscosity tensor in a canonical separated flow: what is the upper bound of accuracy for local Reynolds stress models?, in: AIAA SciTech 2022 Forum, 2022, p. 0940.
[14] Z. Li, N.B. Kovachki, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, in: International Conference on Learning Representations, 2020, pp. 1–16.
[15] L. Lu, P. Jin, G. Pang, Z. Zhang, G.E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, Nat. Mach. Intell. 3 (2021) 218–229.
[16] R.H. Kraichnan, Eddy viscosity and diffusivity: exact formulas and approximations, Complex Syst. 1 (1987) 805–820.
[17] J. Liu, H. Williams, A. Mani, A systematic approach for obtaining and modeling a nonlocal eddy diffusivity, arXiv:2111.03914, 2021.
[18] Y. Shirian, A. Mani, Eddy diffusivity operator in homogeneous isotropic turbulence, Phys. Rev. Fluids 7 (2022) L052601.
[19] O.B. Shende, A. Mani, Closures for multicomponent reacting flows based on dispersion analysis, Phys. Rev. Fluids 7 (2022) 093201.
[20] O.B. Shende, A. Mani, A nonlocal extension of dispersion analysis for closures in reactive flows, arXiv:2201.10013, 2022.
[21] D. Park, A. Mani, Direct calculation of the eddy viscosity operator in turbulent channel flow at $Re_\tau = 180$, arXiv:2108.10898, 2021.
[22] R. Altmann, P. Henning, D. Peterseim, Numerical homogenization beyond scale separation, Acta Numer. 30 (2021) 1–86.
[23] F. Hamba, Nonlocal expression for scalar flux in turbulent shear flow, Phys. Fluids 16 (2004) 1493–1508.
[24] L. Lin, J. Lu, L. Ying, Fast construction of hierarchical matrix representation from matrix–vector multiplication, J. Comput. Phys. 230 (2011) 4071–4087.
[25] P.-G. Martinsson, Compressing rank-structured matrices via randomized sampling, SIAM J. Sci. Comput. 38 (2016) A1959–A1986.
[26] J. Levitt, P.-G. Martinsson, Randomized compression of rank-structured matrices accelerated with graph coloring, arXiv:2205.03406, 2022.
[27] J. Levitt, P.-G. Martinsson, Linear-complexity black-box randomized compression of hierarchically block separable matrices, arXiv:2205.02990, 2022.
[28] M.V. de Hoop, N.B. Kovachki, N.H. Nelsen, A.M. Stuart, Convergence rates for learning linear operators from noisy data, SIAM/ASA J. Uncertain. Quantificat. 11 (2023) 480–513.
[29] N. Boullé, A. Townsend, Learning elliptic partial differential equations with randomized linear algebra, Found. Comput. Math. 23 (2022) 1–31.
[30] G. Stepaniants, Learning partial differential equations in reproducing kernel Hilbert spaces, arXiv:2108.11580, 2021.

[31] N.H. Nelsen, A.M. Stuart, The random feature model for input-output maps between Banach spaces, SIAM J. Sci. Comput. 43 (2021) A3212–A3243.
[32] F. Schäfer, T.J. Sullivan, H. Owhadi, Compression, inversion, and approximate PCA of dense kernel matrices at near-linear computational complexity, SIAM J. Multiscale Model. Simul. 19 (2021) 688–730.
[33] F. Hamba, Nonlocal analysis of the Reynolds stress in turbulent shear flow, Phys. Fluids 17 (2005) 115102.
[34] S.H. Bryngelson, K. Schmidmayer, T. Colonius, A quantitative comparison of phase-averaged models for bubbly, cavitating flows, Int. J. Multiph. Flow 115 (2019) 137–143.
[35] S.H. Bryngelson, A. Charalampopoulos, T.P. Sapsis, T. Colonius, A Gaussian moment method and its augmentation via LSTM recurrent neural networks for the statistics of cavitating bubble populations, Int. J. Multiph. Flow 127 (2020) 103262.
[36] A. Vié, H. Pouransari, R. Zamansky, A. Mani, Particle-laden flows forced by the disperse phase: comparison between Lagrangian and Eulerian simulations, Int. J. Multiph. Flow 79 (2016) 144–158.
[37] M. Ma, J. Lu, G. Tryggvason, Using statistical learning to close two-fluid multiphase flow equations for bubbly flows in vertical channels, Int. J. Multiph. Flow 85 (2016) 336–347.
[38] D.L.O.-L. Lavacot, J. Liu, H. Williams, B.E. Morgan, A. Mani, Nonlocality of mean scalar transport in two-dimensional Rayleigh–Taylor instability using the macroscopic forcing method, arXiv:2307.13911, 2023.
[39] F. Zhang, The Schur Complement and Its Applications, vol. 4, Springer Science & Business Media, 2006.
[40] O. Reynolds IV, On the dynamical theory of incompressible viscous fluids and the determination of the criterion, Philos. Trans. R. Soc. Lond. A (1895) 123–164.
[41] R. Berkowicz, L.P. Prahm, Generalization of K theory for turbulent diffusion. Part I: spectral turbulent diffusivity concept, J. Appl. Meteorol. Climatol. 18 (1979) 266–272.
[42] S. Corrsin, Limitations of Gradient Transport Models in Random Walks and in Turbulence, Advances in Geophysics, vol. 18, Elsevier, 1975, pp. 25–60.
[43] F. Schäfer, M. Katzfuss, H. Owhadi, Sparse Cholesky factorization by Kullback–Leibler minimization, SIAM J. Sci. Comput. 43 (2021) A2019–A2046.
[44] N. Halko, P.-G. Martinsson, J.A. Tropp, Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions, SIAM Rev. 53 (2011) 217–288.