This is an example document showing how to make a nice looking paper with different LaTeXcommands and packages

Your Name[a], Spencer H. Bryngelson[b]

[a]*Some other School, Georgia Institute of Technology, Atlanta, GA 30332, USA*

[b]*School of Computational Science & Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA*

Abstract

This is an excerpt of a paper I published. Some parts were removed to make things cleaner. Use this as an example of formatting your papers and figures, and please realize that important parts of the arguments in this paper may be spuriously missing.

Keywords: Keywords; Here

1. Introduction

Well-established equations describe even the most complicated flow physics. Still, full-resolution simulations of them stretch computational resources. 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, including the RANS closure models [1]. However, data-driven approaches are emerging as semi-automated tools to accomplish the same task. 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 [2, 3]. An alternative approach is to compute effective equations that act on spatial or temporal averages. 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 [4] and Hamba [5] 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 [6] quantifies closure operators exactly by examining forcing and averaged responses, called input–output pairs. However, as a linear-algebra-based technique, MFM does not require the use Dirac delta functions as forcing basis functions, and others like polynomials [7] and harmonic functions [8], can be used.

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 [6] and reconstruct the corresponding eddy diffusivity operator, and momentum transport in a canonical turbulent 3D channel flow at $\mathrm{Re}_\tau = 180$ following Hamba [9] and Park and Mani [10] 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 [11–14], though we do not address such extensions here.

Email address: shb@gatech.edu (Spencer H. Bryngelson)

We briefly introduce MFM and similar approaches for recovering turbulence closure operators in section 2. Results are presented in section 3, focusing on the 2D and 3D problems analyzed by Mani and Park [6] and Park and Mani [10]. Section 4 discusses the outlook of sparse reconstruction methods like the one presented for other flow problems and PDEs broadly.

2. Background on the Macroscopic Forcing Method (MFM)

2.1. The macroscopic forcing method

Given a set of linear microscopic equations,

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

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

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

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

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

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. However, we point out that the techniques outlined in this work apply to a wider range of possible averaging operations.
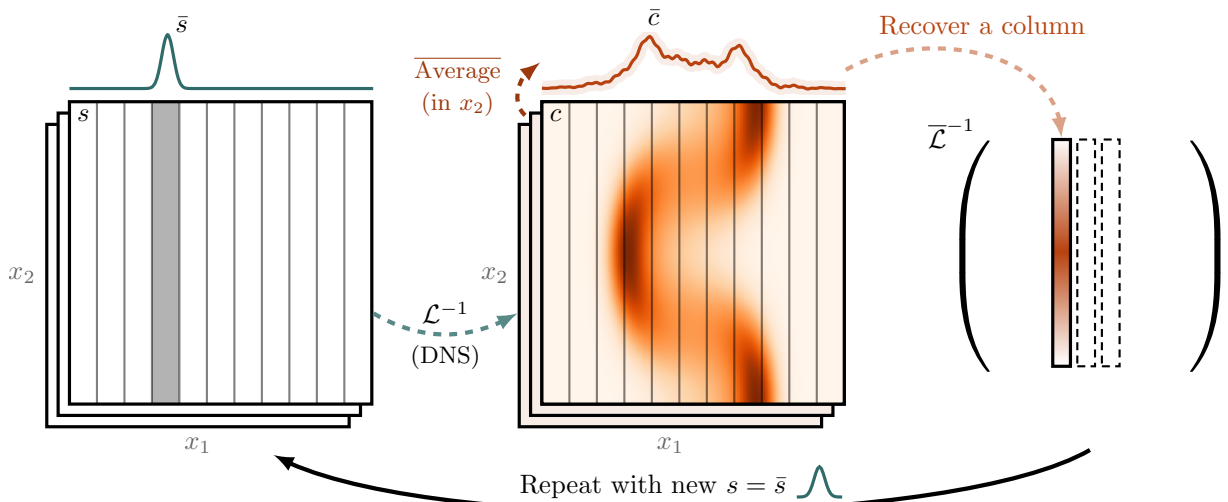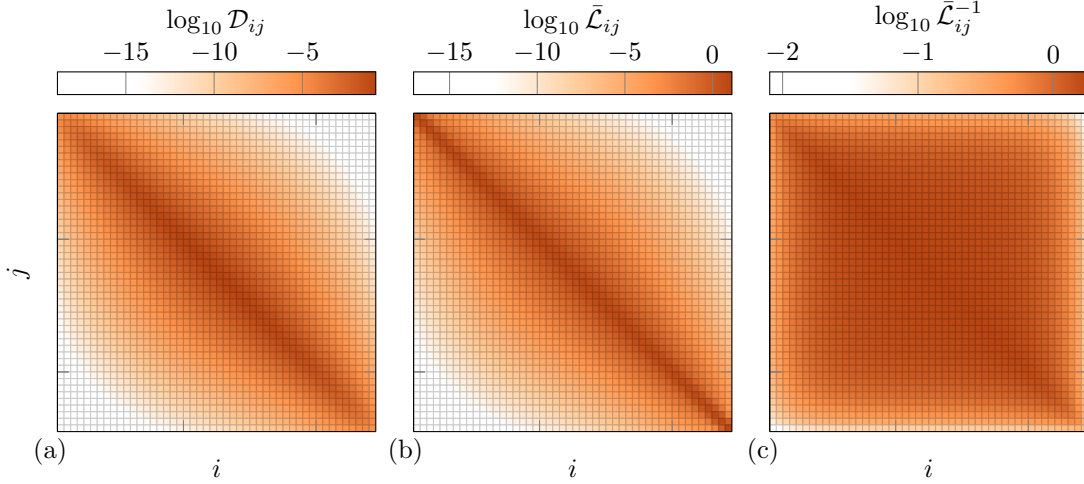

Figure 1: Schematic of the MFM.

Figure 2: Matrices $\mathcal{D}$, $\overline{\mathcal{L}}$, 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.

Using MFM, one can determine the exact linear operator $\overline{\mathcal{L}}$ that acts on averages of flow statistics [6]. They infer this operator by generating solution pairs to $\overline{\mathcal{L}}\bar{c} = \bar{s}$, obtained from solving the microscopic equations with forcing $\bar{s}$ and macroscopic solution average $\bar{c}$.

Figure 1 show 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.

2.2. 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 [15], 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. [16] 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.

2.3. Fast MFM on nonsymmetric operators

As remarked by Schäfer and Owhadi [15], a LU version of Cholesky reconstruction that extends to nonsymmetric matrices requires not only matrix–vector products but also transpose–vector products. Similar requirements arise in hierarchical low-rank approaches [17, 18]. Schur complementation
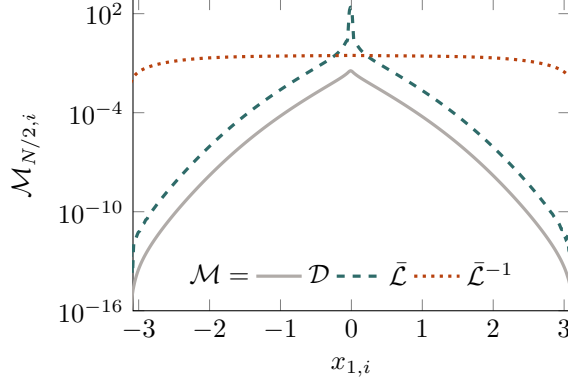
Figure 3: The middle row of the matrices of fig. 2 on a log scale. $\mathcal{D}$ and $\overline{\mathcal{L}}$ have a similar degree of locality, with entries decreasing in magnitude algebraically from the diagonal. The discretized operator $\overline{\mathcal{L}}^{-1}$ is dense; along its diagonal entries only decay modestly at the boundaries.

commutes with transposition, in the sense that

$$\left(\overline{\mathcal{L}}\right)^{\top} = \left(\left(\mathcal{L}^{-\top}\right)_{1,1}\right)^{-1} = \left(\mathcal{L}^{\top}\right)_{1,1} - \left(\mathcal{L}^{\top}\right)_{1,2}\left(\left(\mathcal{L}^{\top}\right)_{2,2}\right)^{-1}\left(\mathcal{L}^{\top}\right)_{2,1}. \tag{5}$$

As a result, transpose–vector products with $\overline{\mathcal{L}}$ can be obtained by applying (inverse or forward) MFM to the $\overline{\mathcal{L}}^{\top}$. 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.

3. Results

3.1. 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 with $x_2 = 0, 2\pi$ 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

$$u_1\frac{\partial c}{\partial x_1} + u_2\frac{\partial c}{\partial x_2} = 0.05\frac{\partial^2 c}{\partial x_1^2} + \frac{\partial^2 c}{\partial x_2^2} + 1, \tag{6}$$

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

$$u_1 = (1 + \cos(2x_1))\cos(2x_2), \quad u_2 = \sin(2x_1)\sin(2x_2). \tag{7}$$

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
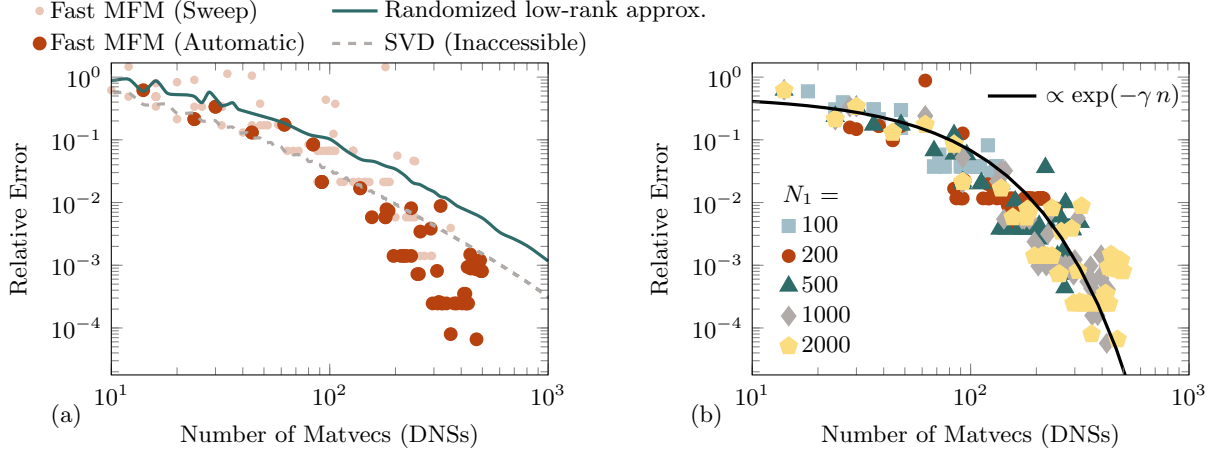
4

Figure 4: Relative L_2 errors between the exact MFM operator $\mathcal{D}$ 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.

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.

Figure 4 shows the macroscopic 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 $\mathcal{D}$ but requires access to the full operator and is, therefore, not practical. A randomized low-rank representation is also shown, which is available. 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 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 still produces poor parameter choices resulting in larger Fast MFM errors.

In fig. 4 (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 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.

Figure 5 shows the application of the recovered operator of fig. 4 to compute $\bar{c}$. 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
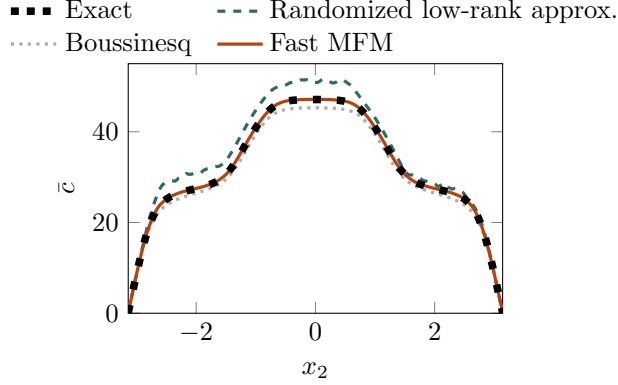
Figure 5: Reconstruction of $\bar{c}$ for the laminar, steady channel flow problem. Results are shown for a randomized low-rank approximation, the Boussinesq approximation, and the Fast MFM result. This visualization is 26 out of 2000 possible samples.

not. The Boussinesq approximation is purely local [19]:

$$-\overline{u_1' c'} = D_{\text{Boussinesq}} \frac{\partial \bar{c}}{\partial x_1}, \tag{8}$$

where

$$D_{\text{Boussinesq}}(x_1) = \int_{y_1} \mathcal{D}(x_1, y_1) \mathrm{d}y_1. \tag{9}$$

3.2. Turbulent channel flow

We next consider a fully-developed turbulent channel flow, reconstructing eddy diffusivities, which, for momentum transport, are commonly referred to as eddy viscosities. The incompressible Navier–Stokes equations are

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_j u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{\text{Re}} \frac{\partial^2 u_i}{\partial x_j \partial x_j} + r_i, \tag{10}$$

$$\frac{\partial u_j}{\partial x_j} = 0, \tag{11}$$

where r is a body force, p is the pressure, and u are velocities. Following Mani and Park [6], the generalized momentum transport equation associated with MFM for a computed u_i field is

$$\frac{\partial v_i}{\partial t} + \frac{\partial u_j v_i}{\partial x_j} = -\frac{\partial q}{\partial x_i} + \frac{1}{\text{Re}} \frac{\partial^2 v_i}{\partial x_j \partial x_j} + s_i, \tag{12}$$

$$\frac{\partial v_j}{\partial x_j} = 0, \tag{13}$$

for a pressure-like term q that ensures incompressibility.

We consider a case with $\text{Re}_\tau = u_\tau \delta / \nu = 180$ 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, r, is the mean pressure gradient in the periodic simulation and is
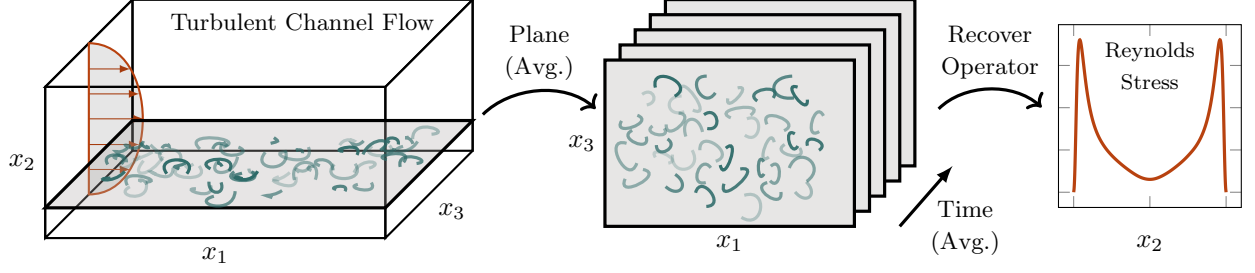
Figure 6: Diagram of the Fast MFM reconstruction procedure for the 3D turbulent channel flow case.
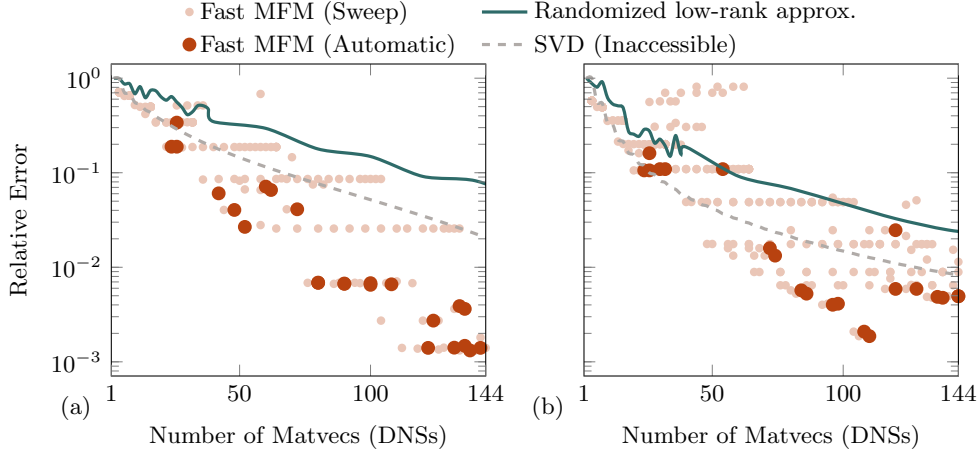

Figure 7: Relative errors in the recovered eddy viscosity kernels (a) $\mathcal{D}^{21}$ and (b) $\mathcal{D}^{11}$ are shown for the $\mathrm{Re}_\tau = 180$ turbulent channel flow configuration.

$\boldsymbol{r} = (1, 0, 0)$ in this nondimensionalized problem. The incompressible Navier–Stokes equations are solved using direct numerical simulation with a 144^3 grid for $T = 500\delta/u_\tau$ to ensure statistical convergence. Simulation-result baselines, including the discretized $\mathcal{D}$ and $\mathcal{L}$ matrices, for this case follow from Park and Mani [10] and are used herein. Figure 6 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. The generalized eddy viscosity is given by

$$-\overline{u'_\beta u'_\gamma}(\boldsymbol{x}) = \int_{\boldsymbol{y}} \mathcal{D}^{(\beta\gamma)}(\boldsymbol{x}, \boldsymbol{y}) \frac{\partial \bar{u}_1}{\partial x_2}\bigg|_{\boldsymbol{y}} \, \mathrm{d}\boldsymbol{y} \tag{14}$$

where $\mathcal{D}^{\beta\gamma}$ in our notation represents $\mathcal{D}_{\beta\gamma 21}$ in traditional notation for the eddy viscosity tensor [9].

Figure 7 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 IMFM, which computes each non-averaged degree of freedom via forcing each s_i independently to recover all columns of $\mathcal{D}^{\beta\gamma}$ for each β and γ. In fig. 7, 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 increases. Fast MFM has a factor of about 100 smaller errors than the low-rank
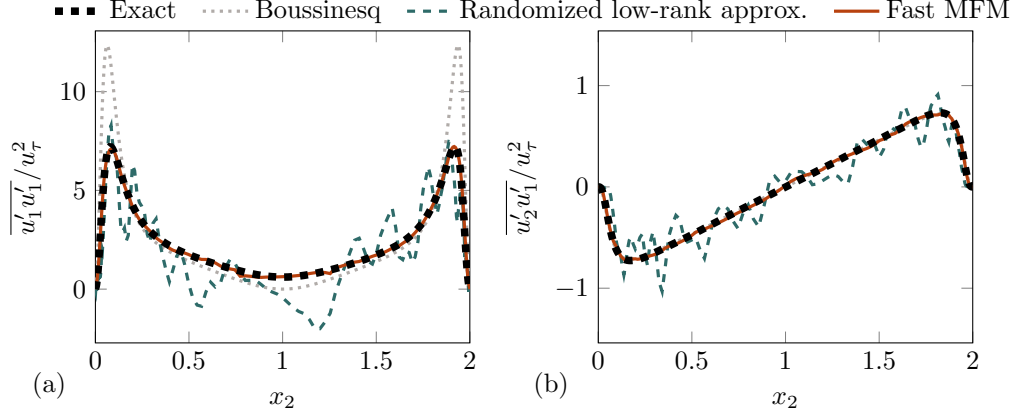
Figure 8: Reynolds stress reconstructions (a) $\overline{u_1' u_1'}$ and (b) $\overline{u_2' u_1'}$ for the turbulent channel flow configuration. Results are shown for 20 out of 144 possible samples.

approximation for 100 matrix–vector products in both (a) and (b).

Figure 8 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 Boussinesq approximation 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 Boussinesq approximation is a good one for the transverse stress component $\overline{u_2' u_1'}$ of fig. 8 (b) but does poorly with the $\overline{u_1' u_1'}$ reconstruction in fig. 8 (a).

4. Conclusion

This work explores a linear algebra approach to reconstructing 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. While other ongoing work focuses on modeling the nonlocal eddy diffusivity using partial differential equations and limited information about the exact eddy diffusivity, 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.

Acknowledgements

References

[1] H. Tennekes, J. L. Lumley, A first course in turbulence, MIT press, 1972.

[2] 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.

[3] L. Lu, P. Jin, G. Pang, Z. Zhang, G. E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, Nature Machine Intelligence **3** (2021) 218–229.

[4] R. H. Kraichnan, Eddy viscosity and diffusivity: exact formulas and approximations, Complex Systems **1** (1987) 805–820.

[5] F. Hamba, An analysis of nonlocal scalar transport in the convective boundary layer using the green's function, Journal of Atmospheric Sciences **52** (1995) 1084–1095.

[6] A. Mani, D. Park, Macroscopic Forcing Method: A tool for turbulence modeling and analysis of closures, Physical Review Fluids **6** (2021) 054607.

[7] J. Liu, H. Williams, A. Mani, A systematic approach for obtaining and modeling a nonlocal eddy diffusivity, arXiv:2111.03914 (2021).

[8] Y. Shirian, A. Mani, Eddy diffusivity operator in homogeneous isotropic turbulence, Physical Review Fluids **7** (2022) L052601.

[9] F. Hamba, Nonlocal analysis of the Reynolds stress in turbulent shear flow, Physics of Fluids **17** (2005) 115102.

[10] D. Park, A. Mani, Direct calculation of the eddy viscosity operator in turbulent channel flow at $Re_\tau = 180$, arXiv:2108.10898 (2021).

[11] S. H. Bryngelson, K. Schmidmayer, T. Colonius, A quantitative comparison of phase-averaged models for bubbly, cavitating flows, International Journal of Multphase Flow **115** (2019) 137–143.

[12] 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, International Journal of Multiphase Flow **127** (2020) 103262.

[13] A. Vié, H. Pouransari, R. Zamansky, A. Mani, Particle-laden flows forced by the disperse phase: Comparison between Lagrangian and Eulerian simulations, International Journal of Multiphase Flow **79** (2016) 144–158.

[14] M. Ma, J. Lu, G. Tryggvason, Using statistical learning to close two-fluid multiphase flow equations for bubbly flows in vertical channels, International Journal of Multiphase Flow **85** (2016) 336–347.

[15] F. Schäfer, H. Owhadi, Sparse recovery of elliptic solvers from matrix–vector products, arXiv:2110.05351 (2023).

[16] F. Schäfer, T. J. Sullivan, H. Owhadi, Compression, inversion, and approximate PCA of dense kernel matrices at near-linear computational complexity, SIAM Multiscale Modeling & Simulation **19** (2021) 688–730.

[17] N. Halko, P.-G. Martinsson, J. A. Tropp, Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, SIAM Review **53** (2011) 217–288.

[18] L. Lin, J. Lu, L. Ying, Fast construction of hierarchical matrix representation from matrix–vector multiplication, Journal of Computational Physics **230** (2011) 4071–4087.

[19] F. Hamba, Nonlocal expression for scalar flux in turbulent shear flow, Physics of Fluids **16** (2004) 1493–1508.