Center for Turbulence Research
Proceedings of the Summer Program 2022

Fast Macroscopic Forcing Method

By S. H. Bryngelson†, F. Schäfer†, J. Liu AND A. Mani

The macroscopic forcing method (MFM) is a method of Green's function quantification introduced by Mani & Park (2021). It recovers reduced operators by means of repeated direct numerical simulation. The MFM has already successfully recovered Reynolds-averaged Navier–Stokes (RANS)-like operators for turbulent flows. Standard algorithms for the MFM apply forcings to each coarse-scale degree of freedom and conduct a fine-scale simulation, which is expensive. We present an algorithm that is cheaper and more general. The fast MFM we introduce applies sparse reconstruction to expose local features in the coarse-grained differential operator and reconstructs the coarse one in only a few matrix–vector products. For non-local operators, the algorithm first peels long-range effects with dense matrix–vector products to expose a more local operator. We demonstrate the algorithm's performance on laminar and turbulent channel flow problems. For both problems, we recover eddy–diffusivity-like operators associated with RANS closure models. We observe that we can reconstruct the eddy diffusivity operator with an increase in accuracy by about a factor of 100 over randomized low-rank methods. Applying this operator to compute the averaged fields of interest has visually indistinguishable behavior from the exact solution, but entails 1% of the cost of computing the exact operator via the MFM of Mani & Park (2021) for a laminar channel flow and 13% for a turbulent one. Our results show that a similar number of simulations are required to reconstruct the operators to the same accuracy under grid refinement, and thus the accuracy corresponds to the physics of the problem, not the numerics. We glean that eddy-diffusivity operators can be reconstructed with reasonable accuracy using only a few simulations, regardless of simulation resolution or degrees of freedom.

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 were sufficient to develop these models, such as the RANS closure. Data-driven approaches are emerging as semi-automated tools to accomplish the same task and are sometimes better, faster, or both. Some approaches attempt to coarse-grain the time evolution of the physical system via neural networks, a formidable task that involves reducing the entire Navier–Stokes operator (Li *et al.* 2020; Lu *et al.* 2021). An alternative approach is to compute effective equations that act on spatial or temporal averages. In the spirit of numerical homogenization (Altmann *et al.* 2021), the macroscopic forcing method (MFM) of Mani & Park (2021) is one example of this. All of the above techniques amount to operator recovery or learning, where an unknown operator is estimated from a set of input–output pairs obtained from full-resolution sim-

† School of Computational Science & Engineering, Georgia Institute of Technology. These authors contributed equally to this work (listed in alphabetical order).

ulations. These simulations are expensive, creating a pressing need to reduce the number of samples needed. We discuss how to accomplish this next.

The MFM constructs effective operators acting on solution averages from simulations of the microscopic equations, which have full resolution and dimensionality (Mani & Park 2021). They use 2D and 3D channel flows as example configurations. In the case of channel flow, the wall-normal velocity is the slow direction and the others are averaged into an approximate unknown operator associated with eddy diffusivity, resulting in the macroscopic equations. The MFM has also been successfully used to obtain closures of reacting flow equations (Shende & Mani 2021) and analyze homogeneous isotropic turbulence (Shirian & Mani 2022).

If the macroscopic operators are linear, the MFM procedure is no different from estimating a matrix from a limited number of matrix–vector products. The number of microscopic simulations required to exactly recover the macroscopic operator is equal to the number of macroscopic degrees of freedom, which can be prohibitively large when one considers the computational burden of large simulations. One can partially address this problem by working in Fourier space (Mani & Park 2021) or fitting a parametric model to moments of the eddy-diffusion operator (Liu *et al.* 2021; Park & Mani 2021). However, the former requires spatial homogeneity and the accuracy of the latter depends on the quality of the parametric model.

For many flows of practical interest, the non-local effects of closure terms show diffusive behavior. Thus, work on operator recovery for elliptic partial differential equations (PDEs) is closely related to the MFM. Lin *et al.* (2011) propose a "peeling" approach for recovering hierarchical matrices from a polylogarithmic number of matrix–vector products, although without rigorous bounds on the approximation error. Eigendecompositions and randomized linear algebra have been used to recover elliptic solution operators from matrix–vector products, with rigorous error bounds (de Hoop *et al.* 2021; Boullé & Townsend 2022). However, these methods require poly$(1/\varepsilon)$ matrix–vector products to obtain an ε approximation of the operator. Schäfer *et al.* (2017) leverage the sparse Cholesky factors of elliptic solution operators to provably obtain ε approximations from only poly$(\log(1/\varepsilon))$ matrix–vector products.

We use this approach to craft an accelerated MFM we call the fast MFM. It uses the locality of the physical models to reduce the sample complexity of standard MFM operator recovery. We apply the fast MFM to canonical 2D and 3D channel flow problems that are inhomogeneous and turbulent, then reconstruct the operator associated with the eddy diffusivity. These examples display sufficiently rich spatiotemporal dynamics to argue that the fast MFM could be applied broadly, such as to the closure problems of compressible multiphase flows (Vié *et al.* 2016; Bryngelson *et al.* 2019).

We give a brief background of the MFM in Section 2. Section 3 details the mathematical foundations of the sparse reconstruction procedure and applies it to the MFM, resulting in the fast macroscopic forcing method. Results are presented in Section 4, with focus on the 2D and 3D problems analyzed by Mani & Park (2021). Section 5 discusses the outlook of sparse reconstruction methods like the one presented for other flow problems and PDEs broadly.

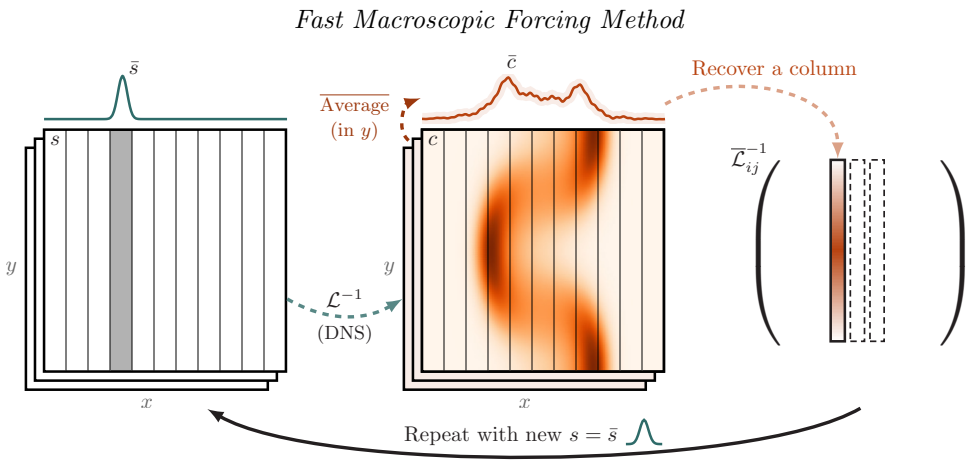

FIGURE 1. Schematic of the MFM.

2. Background on the Macroscopic Forcing Method (MFM)

2.1. *The macroscopic forcing method*

Given a set of linear microscopic equations,

$$\mathcal{L}c = s, \tag{2.1}$$

and an averaging operator $(s, c) \mapsto (\bar{s}, \bar{c})$, the macroscopic (averaged) operator $\bar{\mathcal{L}}$ is defined to satisfy, for all microscopic solutions c, s of Eq. (2.1),

$$\bar{\mathcal{L}}\bar{c} = \bar{s}. \tag{2.2}$$

In the examples presented in this work, averaging occurs with respect to time and all but one spatial component, over the entire domain. However, the techniques formulated here are applicable to a wider range of averaging operations.

The goal of the MFM is to approximate a linear operator $\bar{\mathcal{L}}$ that acts on averages of flow statistics (Mani & Park 2021). These researchers infer this operator by generating solution pairs to $\bar{\mathcal{L}}\bar{c} = \bar{s}$, obtained from solving the microscopic equations with forcing $\bar{s}$ and macroscopic solution average $\bar{c}$. Figure 1 shows an example MFM procedure schematically for a two-dimensional problem with coordinate directions x and y. In it, the relevant averaging direction is y, with averaged "strips" indicating averaging. This solution is forced by a field $s(x, y)$ as a Dirac delta function at an x coordinate that is equivalent to its averaged field $\bar{s}(x)$. The inverse solution operator $\mathcal{L}^{-1}$ solves Eq. (2.1) for $c(x, y)$ given $\bar{s}(x)$. This is equivalent to solving the full-resolution system in Eq. (2.1), or direct numerical simulation (DNS) in the case of turbulent flows. The averaged solution field $\bar{c}(x)$ corresponds to a column of a macroscopic solution operator $\bar{\mathcal{L}}^{-1}$ under this scheme. This procedure is repeated for all non-averaged degrees of freedom.

In this example, the non-averaged coordinate is x, and so each discretized x_i is forced by a Dirac delta at that location with $s \equiv \bar{s}$. Completing the MFM gives access to the matrix representation $\mathbf{L}$ of $\bar{\mathcal{L}}$ via $\bar{s} \mapsto \mathbf{L}^{-1}\bar{s}$. Since evaluation of this map involves a high-resolution simulation, column-by-column construction of $\mathbf{L}^{-1}$ is intractable.

For spatially homogeneous flows, the Fourier transform can diagonalize the matrix $\mathbf{L}$ and recover it from a single matrix–vector product with an all-ones vector. However, the spatial homogeneity assumption does not apply broadly. Mani & Park (2021) indeed address a spatially inhomogeneous case, expressing the macroscopic solution operator

in terms of an eddy diffusivity. Using carefully chosen right-hand sides, they compute moments of the integral kernel associated with the non-local eddy-diffusivity operator. When combined with a parametric model, they can reconstruct the full operator. The accuracy of this approach depends on the validity of the parametric model itself and the smoothness of the eddy-diffusivity kernel.

2.2. *The linear algebra of the MFM*

A linear algebraic perspective helps to understand the MFM. To this end we denote as $\mathcal{L}$ the matrix representation of a discretized advection–diffusion operator

$$\mathcal{L} = \frac{\partial}{\partial t} + v \cdot \nabla - \nabla \cdot (a \nabla), \tag{2.3}$$

where coefficients a, v, are allowed to vary in both space and time. The inverse $\mathcal{L}^{-1}$ of $\mathcal{L}$ takes a spatiotemporal forcing term f as input and returns the spatiotemporal field $u = \mathcal{L}^{-1} f$ solving the PDE. Let $\mathbf{P}$ and $\mathbf{E}$ denote projections onto a set of coarse-scale features of interest, usually spatiotemporal averages. In the example of Figure 1, rows of $\mathbf{P}$ and columns of $\mathbf{E}$ correspond to averages of c, respectively, in the y-direction of the domain. The MFM operator can then be expressed as

$$\bar{\mathcal{L}} = \left(\mathbf{P}\mathcal{L}^{-1}\mathbf{E}\right)^{-1}. \tag{2.4}$$

Another perspective of the MFM is obtained by considering $\bar{\mathcal{L}}$ in discretized form. By using bases for its row and column space that comprise the row and column spaces of $\mathbf{P}$ and $\mathbf{E}$, we obtain a 2×2 block matrix. The MFM is then obtained as the Schur complement of $\mathcal{L}$ after eliminating the second block.

$$\bar{\mathcal{L}} = \left(\left(\mathcal{L}^{-1}\right)_{1,1}\right)^{-1} = \mathcal{L}_{1,1} - \mathcal{L}_{1,2}\left(\mathcal{L}_{2,2}\right)^{-1}\mathcal{L}_{2,1}. \tag{2.5}$$

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

2.3. *The inverse MFM*

As shown in Figure 2, $\bar{\mathcal{L}}$ is more local than its computed inverse. Hoping to turn this locality into computational gains, Park & Mani (2021) propose an inverse MFM to directly compute matrix–vector products with $\bar{\mathcal{L}}$ without first having to compute $\bar{\mathcal{L}}^{-1}$ using the ordinary MFM. This procedure can be interpreted as evaluating the right-hand side of Eq. (2.5) at the cost of solving a system of equations in $\mathcal{L}_{2,2}$. If Eq. (2.1) is an evolution PDE, this procedure can be interpreted as a control problem, where at each time step, the microscopic portion of the forcing s is chosen to maintain a target average $\bar{c}$. The resulting averaged forcing $\bar{s}$ is then equal to $\bar{\mathcal{L}}\bar{c}$.

2.4. *The eddy-diffusivity operator*

Throughout the work of Mani & Park (2021), the averaging operation includes averaging over the entire temporal domain, and thus $\bar{\mathcal{L}}$ can be reformulated as a spatial PDE operator. In this setting, Mani & Park (2021) observe that $\bar{\mathcal{L}}$ can be written as

$$\bar{\mathcal{L}} = -\left(\frac{\partial}{\partial x_N}\right)\left(\mathcal{D} + \mu\mathcal{I}\right)\left(\frac{\partial}{\partial x_N}\right), \tag{2.6}$$

where $\mathcal{D}$ is the eddy-diffusivity matrix, μ is molecular viscosity, and x_N is the non-averaged direction. As shown in Figure 2, the eddy-diffusivity matrix is considerably

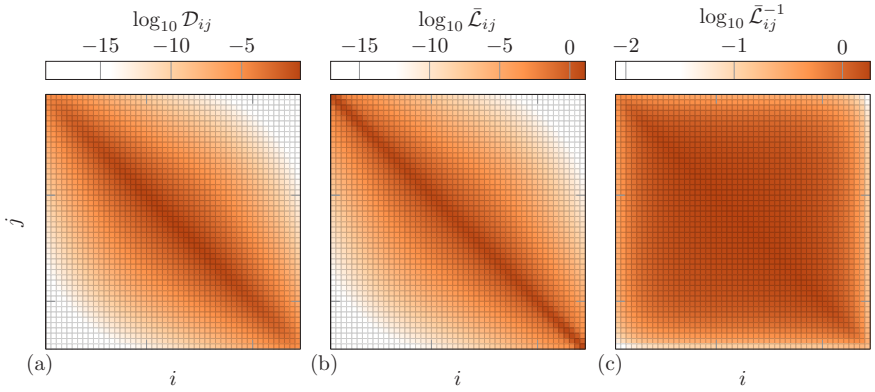

FIGURE 2. Matrices $\mathcal{D}$, $\bar{\mathcal{L}}$, and $\bar{\mathcal{L}}^{-1}$ as labeled for the 2D channel flow case with 50 non-averaged grid points (and so each is 50×50) for illustration purposes. In (a) and (b) we observe a similar degree of bandedness, though in the inverse operator of (c) the matrix appears nearly dense. A similar behavior is observed for finer discretizations, which correspond to larger matrices.

smoother than $\bar{\mathcal{L}}$ while sharing its decay properties. To leverage these properties in operator recovery, Mani & Park (2021) propose a method for computing matrix–vector products $\mathcal{D}\mathbf{v}$ as $\Sigma\bar{\mathcal{L}}\Sigma\mathbf{v}$, where Σ denotes the antiderivative. The objective of the present work is to recover $\mathcal{D}$, and thus $\bar{\mathcal{L}}$, from as few matrix–vector products as possible, as accurately as possible.

3. Lower–upper (LU) reconstruction of elliptic operators

3.1. *Reconstructing elliptic operators from matrix–vector products*

We construct the eddy-diffusivity operator using a variant of the Cholesky reconstruction of Schäfer & Owhadi (2021). This method provably recovers solution operators of divergence form elliptic PDEs in dimension d to accuracy ϵ from only $\mathcal{O}\left(\log^{d+1}(\epsilon^{-1})\right)$ solutions for carefully selected forcing terms. We now briefly review Cholesky reconstruction.

3.2. *Graph coloring*

Graph coloring allows reconstruction of 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 Figure 3, the selected columns can be read off from the resulting matrix–vector product. In a similar manner, graph coloring can also be used to reveal the leading columns of a sparse LU or Cholesky factorization. Once a row-column-pair of the LU factors is 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.* (2011), is referred to as peeling (Figure 3).

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.* (2017) show that when represented in a multiresolution basis ordered from coarse to fine, solution operators of elliptic PDEs have almost sparse Cholesky factors. The leading columns of the Cholesky

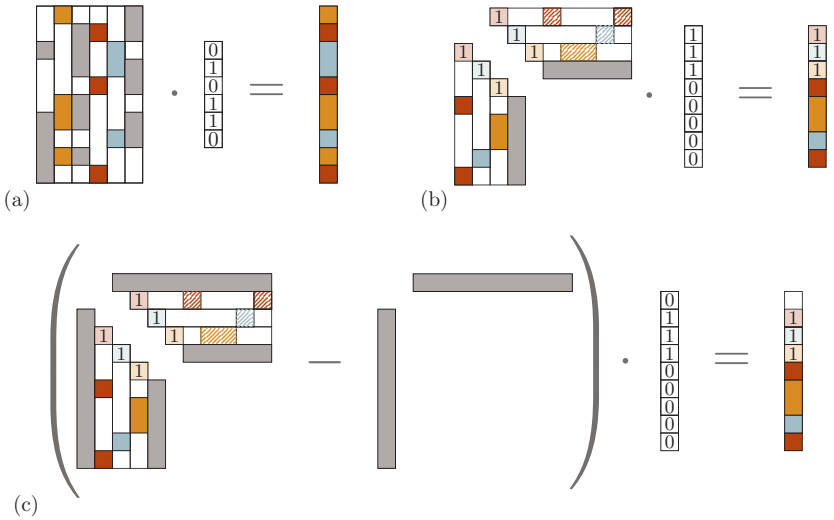

FIGURE 3. (a) Sparse recovery: Columns with non-overlapping but known sparsity patterns (shown in color) can be recovered by a single matrix–vector product via a carefully chosen vector. (b) Factorization: Cholesky factorizations with leading-column sparsity patterns can be recovered in the same way. (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. [This figure was adapted from Schäfer & Owhadi (2021) with author permission.]

factors, corresponding to coarse-scale basis function with global support, are dense and therefore limit the efficiency of graph coloring. However, they are few in number and can be identified efficiently and then removed via peeling. This procedure can be repeated to reveal 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.

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 & Owhadi (2021) recovery, we use an LU recovery that recovers a sparse LU factorization of the target matrix. Columns of L are recovered from matrix–vector products, whereas rows of U are recovered from matrix–transpose–vector products that can be computed by solving the adjoint equation of $\mathcal{L}$. At present, no rigorous guarantees exist for the accuracy of LU reconstruction applied to eddy-diffusivity matrices. However, Schäfer *et al.* (2017) observe empirically that a wide range of diffusion-like operators such as those produced by fractional-order Matérn kernels or Cauchy kernels produce sparse Cholesky factors, despite the lack of theory supporting this observation.

4. Results

4.1. *Steady-state laminar channel flow*

We 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 $x_2 = 0, 2\pi$

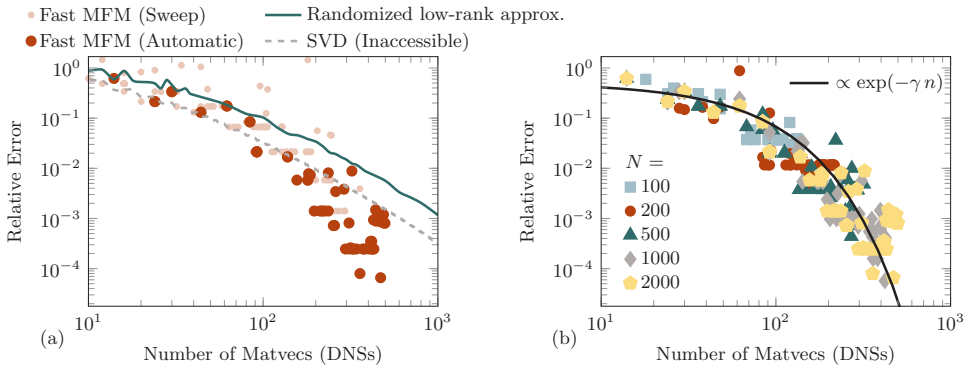

FIGURE 4. Relative L_2 errors between the exact MFM operator $\bar{\mathcal{L}}$, computed via brute-force MFM, and its reconstruction for the laminar flow configuration. In (b), only the fast MFM with automatically chosen parameterization is shown, but for different resolutions N.

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

$$u_1 \frac{\partial c}{\partial x_1} + u_2 \frac{\partial c}{\partial x_2} = 0.05 \frac{\partial^2 c}{\partial x_1^2} + \frac{\partial^2 c}{\partial x_2^2}, \tag{4.1}$$

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

$$u_1 = (1 + \cos(2x_1)) \cos(2x_2), \quad u_2 = \sin(2x_1) \sin(2x_2). \tag{4.2}$$

Figure 4 shows the operator errors for the laminar flow configuration. In Figure 4(a), the mesh resolution in the x_1 (non-averaged) coordinate is $N = 2000$. We see fast MFM errors are smaller than those of a truncated singular value decomposition (SVD), which is inaccessible during the MFM procedure. A randomized low-rank representation is also shown, which has errors about a factor of 10 larger than the SVD, and, likewise, has errors larger than the fast MFM technique by several orders of magnitude as the number of matrix–vector products increases. A sweep of the parameterization of the fast MFM peeling operator is also shown. We use heuristics to set these parameters, which results in the darker marks, though in principle poor parameterizations can result in larger fast MFM errors. In Figure 4(b), we show a similar analysis for the fast MFM results at different mesh resolutions N. Errors decrease exponentially with the number of matrix–vector products with about the same fit coefficients regardless of N. This indicates that the fast MFM reconstruction is dependent on the physical locality of the operator, not a numerical or discretized one, and thus operator recovery for high-resolution simulations has an outsized benefit over the traditional MFM.

Figure 5 shows the application of the recovered operator of Figure 4 to compute $\bar{c}$. The exact result is computed by computing $N = 2000$ direct numerical simulations to recover $\bar{c}$. For fewer direct numerical simulations, 52 out of the 2000 total non-averaged degrees of freedom, the fast MFM method matches the exact result well but the other methods do not.

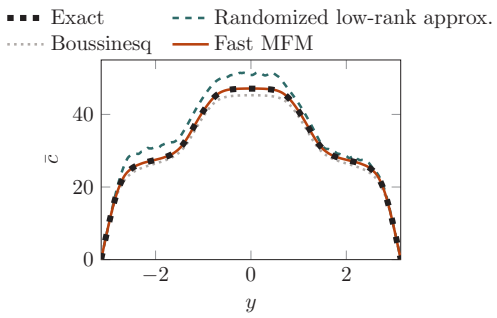

FIGURE 5. Reconstruction of $\bar{c}$ for the laminar, steady channel flow problem, using a randomized low-rank approximation, the Boussinesq approximation, and the present fast MFM method.

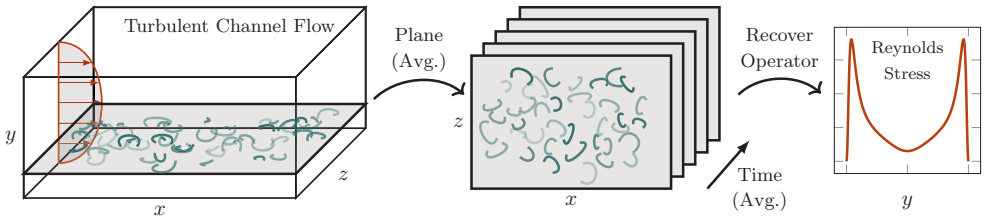

FIGURE 6. Diagram of the fast MFM reconstruction for the 3D turbulent channel flow case.

4.2. *Turbulent channel flow*

We next consider a fully developed incompressible turbulent channel flow, reconstructing eddy diffusivities. Following Mani & Park (2021), the generalized momentum transport equation associated with the MFM for a computed u_i field is

$$\frac{\partial v_i}{\partial t} + \frac{\partial u_j v_i}{\partial x_j} = -\frac{\partial q}{\partial x_i} + \nu \frac{\partial^2 v_i}{\partial x_j \partial x_j} + s_i \quad \text{and} \quad \frac{\partial v_j}{\partial x_j} = 0, \tag{4.3}$$

for a pressure-like term q that ensures incompressibility.

We consider a case with $Re_\tau = u_\tau \delta / \nu = 180$ where δ is the channel half-width. The mean flow is in the x direction, the y direction is wall normal, and the z direction is the span with periodic boundaries. The incompressible Navier–Stokes equations are solved with a 144^3 grid for 950 eddy turnover times to ensure statistical convergence. Figure 6 shows the MFM procedure we use, averaging all independent variables except for the wall-normal coordinate y.

Figure 7 shows the errors in the recovered eddy diffusivities. 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 MFM, which computes each non-averaged degree of freedom via forcing each s_i independently to recover all columns of $\bar{\mathcal{L}}^{-1}$, then computing $\mathcal{D}_{ji}$ via Eq. (2.6) for each j. The trends of Figure 7(a,b) are similar, with the fast MFM method having smaller errors than both the SVD and a randomized low-rank approximation of it. The differences in errors are small for small numbers of matrix–vector produces as the peeling procedure is removing the long-range behaviors. For larger numbers of matrix–vector products, the difference increases. The 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).

Figure 8 show the Reynolds stress reconstructions for the turbulent channel flow. The

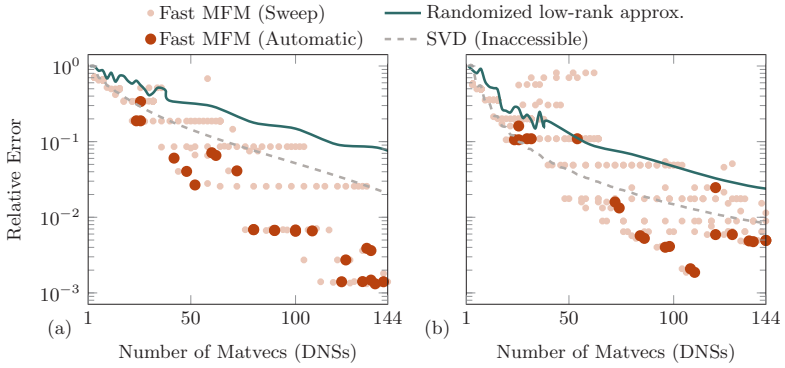

FIGURE 7. Relative errors in the recovered eddy diffusivity kernels (a) $\mathcal{D}_{21}$ and (b) $\mathcal{D}_{11}$ are shown for the $Re_\tau = 180$ turbulent channel flow configuration.

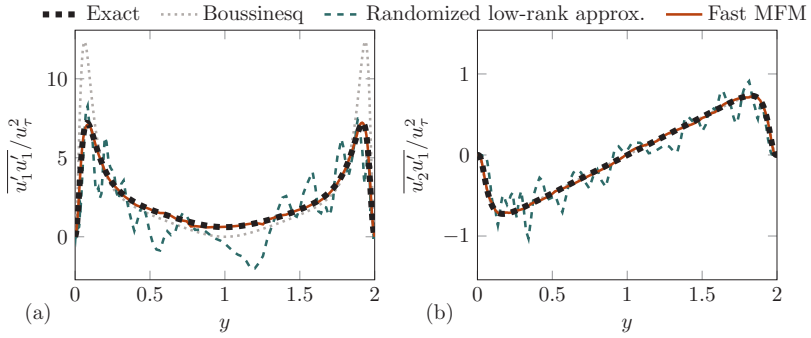

FIGURE 8. Reynolds stress reconstructions (a) $\overline{u_1'u_1'}$ and (b) $\overline{u_2'u_1'}$ for the turbulent channel flow configuration for 20 matrix–vector products (simulations).

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 the transverse stress component $\overline{u_2'u_1'}$ of Figure 8(b) but does poorly with the $\overline{u_1'u_1'}$ reconstruction in Figure 8(a).

5. Conclusion

This work explores a linear algebra approach to reconstructing closure operators. The fast MFM uses sparse recovery and peeling techniques, revealing local behaviors that can be simultaneously recovered by crafted forcings. Results show that using 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 performs poorly in some test cases; randomized low-rank approximations, which while feasible have poor accuracy; and even the SVD, which performs worse than the fast MFM and is inaccessible in a simulation environment. The fast MFM procedure recovers

effective operators at low sample complexity and serves as a stepping stone toward the long-term goal of sample-efficient recovery of coarse-grained time integrators.

Acknowledgements

SHB acknowledges use of the Extreme Science and Engineering Discovery Environment (XSEDE) under allocation TG-PHY210084, which is supported by NSF No. ACI-1548562.

REFERENCES

Altmann, R., Henning, P. & Peterseim, D. 2021 Numerical homogenization beyond scale separation. *Acta Numer.* **30**, 1–86.

Boullé, N. & Townsend, A. 2022 Learning elliptic partial differential equations with randomized linear algebra. *Found. Comput. Math.* pp. 1–31.

Bryngelson, S. H., Schmidmayer, K. & Colonius, T. 2019 A quantitative comparison of phase-averaged models for bubbly, cavitating flows. *Int. J. Multiph. Flow* **115**, 137–143.

de Hoop, M. V., Kovachki, N. B., Nelsen, N. H. & Stuart, A. M. 2021 Convergence rates for learning linear operators from noisy data. arXiv:2108.12515.

Li, Z., Kovachki, N. B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A. & Anandkumar, A. 2020 Fourier Neural Operator for parametric partial differential equations. In *International Conference on Learning Representations*, pp. 1–16.

Lin, L., Lu, J. & Ying, L. 2011 Fast construction of hierarchical matrix representation from matrix–vector multiplication. *J. Comput. Phys.* **230** (10), 4071–4087.

Liu, J., Williams, H. & Mani, A. 2021 A systematic approach for obtaining and modeling a nonlocal eddy diffusivity. arXiv:2111.03914.

Lu, L., Jin, P., Pang, G., Zhang, Z. & Karniadakis, G. E. 2021 Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat. Mach. Intell.* **3** (3), 218–229.

Mani, A. & Park, D. 2021 Macroscopic forcing method: A tool for turbulence modeling and analysis of closures. *Phys. Rev. Fluids* **6**, 054607.

Park, D. & Mani, A. 2021 Direct calculation of the eddy viscosity operator in turbulent channel flow at $\mathrm{Re}_\tau = 180$. arXiv:2108.10898.

Schäfer, F. & Owhadi, H. 2021 Sparse recovery of elliptic solvers from matrix–vector products. arXiv:2110.05351.

Schäfer, F., Sullivan, T. J. & Owhadi, H. 2017 Compression, inversion, and approximate PCA of dense kernel matrices at near-linear computational complexity. arXiv:1706.02205.

Shende, O. B. & Mani, A. 2021 Closures for multi-component reacting flows based on dispersion analysis. arXiv:2111.10551.

Shirian, Y. & Mani, A. 2022 Eddy diffusivity operator in homogeneous isotropic turbulence. *Phys. Rev. Fluids* **7**, L052601.

Vié, A., Pouransari, H., Zamansky, R. & Mani, A. 2016 Particle-laden flows forced by the disperse phase: Comparison between Lagrangian and Eulerian simulations. *Int. J. Multiph. Flow* **79**, 144–158.