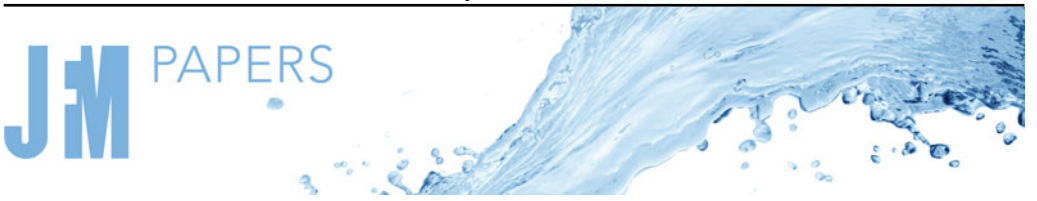
J. Fluid Mech. (2026), *vol.* 1030, A21, doi:10.1017/jfm.2026.11251



Forecasting the evolution of three-dimensional turbulent recirculating flows from sparse sensor data

Shengqi Lu[1] **and George Papadakis**[1]

[1]Department of Aeronautics, Imperial College London, Exhibition Road, London SW7 2AZ, UK

Corresponding author: George Papadakis, g.papadakis@imperial.ac.uk

(Received 8 May 2025; revised 24 January 2026; accepted 26 January 2026)

A data-driven algorithm is proposed that employs sparse data from velocity and/or scalar sensors to forecast the future evolution of three-dimensional turbulent flows. The algorithm combines time-delayed embedding together with Koopman theory and linear optimal estimation theory. It consists of three steps: dimensionality reduction, currently with proper orthogonal decomposition (POD); construction of a linear dynamical system for current and future POD coefficients; and system closure using sparse sensor measurements. In essence, the algorithm establishes a mapping from current sparse data to the future state of the dominant structures of the flow over a specified time window. The method is scalable (i.e. applicable to very large systems), physically interpretable and provides sequential forecasting on a sliding time window of prespecified length. It is applied to the turbulent recirculating flow over a surface-mounted cube (with more than 10^8 degrees of freedom) and is able to forecast accurately the future evolution of the most dominant structures over a time window at least two orders of magnitude larger that the (estimated) Lyapunov time scale of the flow. Most importantly, increasing the size of the forecasting window only slightly reduces the accuracy of the estimated future states.

Key words: low-dimensional models, shear layer turbulence, computational methods

1. Introduction

Flow reconstruction from sparse data has received a lot of attention and the literature on the subject is vast; see for example Brunton & Noack (2015), Sipp & Schmid (2016), Callaham, Maeda & Brunton (2019), Karniadakis *et al.* (2021), Zaki (2025) and references therein. On the other hand, forecasting the future evolution of the flow has received

© The Author(s), 2026. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

considerably less attention in fluid dynamics research, especially turbulence. The ability to estimate the future state from current data, also known as forecasting (Box *et al.* 2015), has many applications. Examples include weather forecasting, urban safety (prediction of toxic pollutant trajectory that can guide evacuation measures), prediction of extreme events (such as heat waves) ahead of time, etc.

The forecasting time window of turbulent flows is limited by a fundamental physical constraint, which arises from the fact that such flows are chaotic and thus have extreme sensitivity to initial conditions. This is popularly known as the 'butterfly effect' (Lorenz 1963) and can be understood as follows. Consider two turbulent flows, originating from the same physical process, that have initial conditions with infinitesimally small differences. Their trajectories in phase space will diverge at a rate determined by the maximum Lyapunov exponent, λ_1, that is $|\delta x(t)| \sim \mathrm{e}^{\lambda_1 t}$, where $|\delta x(t)|$ is the norm of the distance between two points in phase space. The corresponding time scale is $1/\lambda_1$.

There are theoretical arguments (Ruelle 1979; Crisanti *et al.* 1993; Ge, Rolland & Vassilicos 2023) and computational evidence (Mohan, Fitzsimmons & Moser 2017; Hassanaly & Raman 2019) that in turbulent flows λ_1 scales with the Kolmogorov time scale, τ (with a correction factor that depends on Reynolds number). Therefore the future evolution of turbulent flows can only be forecast over a time window that is only several times larger than τ, say $(6-10)\tau$. This has been confirmed repeatedly, irrespective of the method used for forecasting; see Pathak *et al.* (2017), Vlachas *et al.* (2020), Dubois *et al.* (2020), Eivazi *et al.* (2021), Khodkar & Hassanzadeh (2021) and Constante-Amores, Linot & Graham (2025) for a very small sample of extensive literature on the subject. Therefore, for high-Reynolds-number flows this time window is very short to be useful in practice. For other applications, however, for example in medium-range weather forecasting, current lead times are 10 days, but can reach up to 15 days if the uncertainty in the initial conditions is reduced by an order of magnitude (Zhang *et al.* 2019; Allen, Markou & Tebbutt 2025). This window is sufficiently large to bring about enormous socioeconomic benefits, for example protecting lives, property, etc.

Is it possible to enlarge the forecasting window for turbulent flows and get estimations that are useful in engineering practice? The aforementioned scaling of λ_1 with τ indicates that the former is determined by the smallest scales of the flow. However, a turbulent flow consists of a wide range of spatio-temporal structures, and it is well known that large-scale structures are slower and more organised (Pope 2000). This opens the possibility that they can be more amenable to forecasting. These structures are important because they determine momentum transfer (and therefore forces) and scalar dispersion. The ability therefore to forecast their future evolution can bring many practical benefits. Most of existing work on forecasting to date has focused on low-dimensional dynamical systems (such the Lorenz 63/96 systems, the Kuramoto–Sivashinsky equation, the nine-equation model of Moehlis, Faisst & Eckhardt (2004)) or two-dimensional flows (such as flow in a two-dimensional lid-driven cavity). These dynamical systems, however, do not exhibit the large separation of length and time scales that is necessary to answer the question at the beginning of this paragraph. To fill this gap, the present paper considers the three-dimensional turbulent recirculating flow around a surface-mounted prism. As is seen later, this flow contains a wide range of scales, such as large organised structures and a clear inertial regime.

Naive application of existing dimensionality reduction methods, such as dynamic mode decomposition (DMD), for forecasting will fail. The DMD method constructs a linear model of the form $\boldsymbol{x}[k+1] = A\boldsymbol{x}[k]$ (see Schmid 2010), where $\boldsymbol{x}[k]$ is the state vector at time instant k. Recursive application of this relation results in exponentially increasing or decreasing state values, depending on whether the eigenvalues of A are outside or inside

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

the unit circle, respectively. In the case of exponentially increasing values, the growth rate is not related to λ_1. The same problem appears for higher-order DMD, which can be put in the same linear form as the standard DMD, but now $\boldsymbol{x}[k]$ contains time-delayed variables (Le Clainche & Vega 2017).

More advanced approaches are therefore necessary. In particular, the decomposition of chaotic dynamics into a linear model with forcing is a very attractive approach because of the availability of a large number of tools for linear systems. Recently Chu & Schmidt (2025) derived a linear, time-invariant model for the coefficients of spectral proper orthogonal decomposition (POD) modes. The authors retained the forcing term and included its dynamics in the model formulation. The presence of a stochastic forcing term results in a probabilistic representation of the future evolution of dominant structures. Brunton *et al.* (2017) combined time-delay embedding and Koopman theory to derive a linear model in the leading time-delay coordinates which is forced by low-energy variables. The authors call this the Hankel alternative view of Koopman (HAVOK). The underlying idea is the following. Koopman theory (Mezić 2013; Otto & Rowley 2021; Brunton *et al.* 2022) seeks a linear representation of strongly nonlinear dynamics in the space of all observables $\boldsymbol{y}$, which are functions of the state $\boldsymbol{x}$ of the system, i.e. $\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x})$. It therefore trades the finite dimensionality of the nonlinear system to the infinite dimensional dynamics of the linear system. In DMD, $\boldsymbol{y} = \boldsymbol{x}$, but this selection of observables may not be rich enough for many nonlinear systems. The main idea of HAVOK is to construct a forced linear model using as states not $\boldsymbol{x}$, but instead time-delayed coordinates that contain more information about the system. This is in accordance with the Takens (1981) embedding theorem; the theorem allows the construction of an attractor from a time series of a single measurement that is diffeomorphic to the original chaotic attractor (diffeomorphic means that a smooth, invertible transformation exists between the original attractor and its representation). A linear model cannot capture multiple fixed points, so the model is forced by an (unclosed) term. The model states are obtained from the right singular vectors of the Hankel matrix which is assembled by stacking time-delayed values of observables column by column, and each column is advanced one time unit ahead of the previous column. It is found that the forcing is active only in the regions of the chaotic attractor with strong nonlinearity (e.g. when switching between the two lobes in the Lorenz attractor), and its statistics are non-Gaussian. In Khodkar & Hassanzadeh (2021) the forcing is found from vector-valued observables in a physics-informed way or a purely data-driven fashion, depending on whether any knowledge of governing dynamics was available or not. In Dylewsky *et al.* (2022) the forcing is obtained in a two-step process in a fully unsupervised manner.

The combination of Koopman theory with time-delayed embeddings has opened new directions for the representation of a chaotic system as a forced linear model and has spawned important theoretical work (Kamb *et al.* 2020; Arbabi & Mezić 2017; Pan & Duraisamy 2019, 2020; Giannakis 2019; Das & Giannakis 2019). The connection between the left-singular vectors of the time-delayed Hankel matrix and the space–time POD, space-only POD and spectral POD modes was established in Frame & Towne (2023).

The contribution of the present paper is to add one more piece to the aforementioned time-delayed embedding/Koopman framework. More specifically, we combine this framework with linear optimal estimation theory (Kailath, Hassibi & Sayed 2000) that allows us to derive a mapping from sparse measurements at the current time instant to the velocity field at future time instants. This approach is useful because it provides an estimate of the forcing term and thereby closes the system. Schmidt (2026) proposed a method that does not require a forcing term. This was achieved by leveraging the correlation between hindcast and forecast datasets with the aid of extended POD. To forecast the

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

future evolution of the flow, data are required for the full flow, while in our method, once the mapping has been established, only data from a few sensors are required. A nonlinear mapping that employs machine learning techniques (instead of Koopman theory) was proposed recently for weather forecasting by Allen *et al.* (2025).

Our approach is applied to flow around a surface-mounted cube, as already mentioned. Scalar is released from a source placed upstream of the cube. We derive the mapping from current to future states using not only velocity data but also scalar data; scalar sensors are usually cheaper to acquire. The proposed estimator is scalable, physically interpretable and provides sequential forecasting on a rolling time window as streaming data are coming in. Of particular interest is the quality of forecasting and how it varies with the size of the time window.

The paper is structured as follows. Sections 2 and 3 describe the forecasting methodology from sparse velocity and scalar data, respectively. Results are presented and discussed in § 4. Section 5 summarises the main findings of the paper and outlines some future research directions.

2. Flow forecasting from sparse velocity measurements

In the following, u, v, w (interchangeably used with u_1, u_2, u_3) are the three velocity components in the Cartesian directions x, y, z, respectively. Time averages are designated with angular brackets and fluctuations with a prime; for example $\langle u \rangle$ and u' are the mean and fluctuating velocities, respectively, in the x direction.

We first describe how to forecast the future evolution of a turbulent flow from a set of sparse velocity measurements that record current information. In the following section, we extend the idea to sparse scalar measurements.

The method comprises three steps.

2.1. *Step 1: dimensionality reduction*

To make the method applicable to three-dimensional turbulent flows, we first need to reduce the number of degrees of freedom. In this paper we use POD (Sirovich 1987). This method provides a linear mapping between the velocity field and the POD coefficients; we exploit this linearity later in step 3.

To get the POD modes, the snapshot matrix $\boldsymbol{Y}(\boldsymbol{x}, t_1 : t_K)$ for the velocity fluctuations u', v' and w' is assembled:

$$\boldsymbol{Y}(\boldsymbol{x}, t_1 : t_K) = \begin{bmatrix} u'(\boldsymbol{x}_1, t_1) & u'(\boldsymbol{x}_1, t_2) & \dots & u'(\boldsymbol{x}_1, t_K) \\ \vdots & \vdots & \dots & \vdots \\ u'(\boldsymbol{x}_N, t_1) & u'(\boldsymbol{x}_N, t_2) & \dots & u'(\boldsymbol{x}_N, t_K) \\ v'(\boldsymbol{x}_1, t_1) & v'(\boldsymbol{x}_1, t_2) & \dots & v'(\boldsymbol{x}_1, t_K) \\ \vdots & \vdots & \dots & \vdots \\ v'(\boldsymbol{x}_N, t_1) & v'(\boldsymbol{x}_N, t_2) & \dots & v'(\boldsymbol{x}_{N,t_K}) \\ w'(\boldsymbol{x}_1, t_1) & w'(\boldsymbol{x}_1, t_2) & \dots & w'(\boldsymbol{x}_1, t_K) \\ \vdots & \vdots & \dots & \vdots \\ w'(\boldsymbol{x}_N, t_1) & w'(\boldsymbol{x}_N, t_2) & \dots & w'(\boldsymbol{x}_{N,t_K}) \end{bmatrix}, \tag{2.1}$$

where $\boldsymbol{Y} \in \mathbb{R}^{3N \times K}$, $\boldsymbol{x}_i = [x_i, y_i, z_i]$ $(i = 1, 2, \dots, N)$ is the location vector for the ith spatial location, N is the number of cells and K is the total number of snapshots. The spacing between successive snapshots is Δt. Singular value decomposition (SVD) is

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

performed on the weighted matrix $\mathcal{V}^{1/2}\boldsymbol{Y}$:

$$\mathcal{V}^{1/2}\boldsymbol{Y} = \boldsymbol{U}_Y \boldsymbol{\Sigma}_Y \boldsymbol{V}_Y^{\top}, \tag{2.2}$$

where $\mathcal{V} = diag(V_1, V_2 \ldots V_N, V_1, V_2, \ldots, V_N, V_1, V_2, \ldots, V_N)$ is a diagonal matrix with the cell volumes V_i in the main diagonal, $\boldsymbol{U}_Y \in \mathbb{R}^{3N\times K}$ contains the left singular vectors, $\boldsymbol{\Sigma}_Y \in \mathbb{R}^{K\times K}$ is a diagonal matrix that stores K singular values (we assume $K \ll 3N$) and $\boldsymbol{V}_Y \in \mathbb{R}^{K\times K}$ contains the right singular vectors. The scaled POD eigenmodes $U_{Y,k}(\boldsymbol{x})$ are extracted from the columns of $\boldsymbol{U}_Y$ from

$$U_Y(\boldsymbol{x}) = \mathcal{V}^{-1/2}\boldsymbol{U}_Y. \tag{2.3}$$

The singular values $\sigma_{Y,k}$ are ranked in descending order along the diagonal of matrix $\boldsymbol{\Sigma}_Y$. The energy content of each mode is computed using

$$\lambda_{Y,k} = \frac{\sigma_{Y,k}^2}{K} \quad (k = 1 \ldots K) \tag{2.4}$$

and the time coefficients from

$$\boldsymbol{a}(t) = \boldsymbol{Y}^{\top}\mathcal{V}^{1/2}\boldsymbol{U}_Y. \tag{2.5}$$

The fluctuating velocity field can be written as

$$u_i'(x, y, z, t) = \sum_{k=1}^{K} a_k(t) U_{Y,k}^{(i)}(x, y, z) \approx \sum_{k=1}^{m_u} a_k(t) U_{Y,k}^{(i)}(x, y, z) \quad (i = 1, 2, 3), \tag{2.6}$$

where m_u is the number of retained POD modes, $a_k(t)$ is the time coefficient of the kth POD mode and $U_{Y,k}^{(i)}$ is the kth POD eigenvector of the ith velocity component. This expression can be written in matrix form as

$$\underbrace{\begin{bmatrix} u_1' \\ u_2' \\ u_3' \end{bmatrix}}_{\equiv \boldsymbol{u}'} = \underbrace{\begin{bmatrix} U_{Y,1}^{(1)} & U_{Y,2}^{(1)} & \ldots & U_{Y,m_u}^{(1)} \\ U_{Y,1}^{(2)} & U_{Y,2}^{(2)} & \ldots & U_{Y,m_u}^{(2)} \\ U_{Y,1}^{(3)} & U_{Y,2}^{(3)} & \ldots & U_{Y,m_u}^{(3)} \end{bmatrix}}_{\equiv \boldsymbol{U}_Y} \underbrace{\begin{bmatrix} a_1' \\ a_2' \\ \vdots \\ a_{m_u}' \end{bmatrix}}_{\equiv \boldsymbol{a}} \tag{2.7}$$

or more compactly

$$\boldsymbol{u}'(x, y, z, t) = \boldsymbol{U}_Y(x, y, z)\boldsymbol{a}(t). \tag{2.8}$$

2.2. *Step 2: construction of a dynamical system for current and future POD coefficients*

The next step is the construction of a model for the dynamic evolution of POD coefficients. This model should be able to forecast the future time signal from current information. To this end, we assemble the time-delayed Hankel matrix $\boldsymbol{H}$ that consists of the POD coefficients of the retained m_u velocity modes:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{a}(t_1) & \boldsymbol{a}(t_2) & \ldots & \boldsymbol{a}(t_p) \\ \boldsymbol{a}(t_2) & \boldsymbol{a}(t_3) & \ldots & \boldsymbol{a}(t_{p+1}) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a}(t_q) & \boldsymbol{a}(t_{q+1}) & \ldots & \boldsymbol{a}(t_{K_{train}}) \end{bmatrix}, \tag{2.9}$$

where $\boldsymbol{a}(t_j) = [a_1(t_j) \ldots a_{m_u}(t_j)]^{\top}$ and we use q vectors in each column. The number of columns is $p = K_{train} - q + 1$, where K_{train} is the number of snapshots for the training

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

dataset and $\boldsymbol{H} \in \mathbb{R}^{(m_u \times q)\times p}$. Performing SVD on $\boldsymbol{H}$, we obtain

$$\boldsymbol{H} \approx \boldsymbol{U}_H \boldsymbol{\Sigma}_H \boldsymbol{V}_H^\top, \tag{2.10}$$

where $\boldsymbol{U}_H \in \mathbb{R}^{(m_u \times q)\times r}$, $\boldsymbol{\Sigma}_H \in \mathbb{R}^{r\times r}$, $\boldsymbol{V}_H \in \mathbb{R}^{p\times r}$ and r is the number of retained singular values. The matrix of the left singular vectors $\boldsymbol{U}_H$ can be explicitly written as

$$\boldsymbol{U}_H = \begin{bmatrix} \boldsymbol{U}_{H,1}^{(u,v,w)}(t_1) & \boldsymbol{U}_{H,2}^{(u,v,w)}(t_1) & \dots & \boldsymbol{U}_{H,r}^{(u,v,w)}(t_1) \\ \boldsymbol{U}_{H,1}^{(u,v,w)}(t_2) & \boldsymbol{U}_{H,2}^{(u,v,w)}(t_2) & \dots & \boldsymbol{U}_{H,r}^{(u,v,w)}(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{U}_{H,1}^{(u,v,w)}(t_q) & \boldsymbol{U}_{H,2}^{(u,v,w)}(t_q) & \dots & \boldsymbol{U}_{H,r}^{(u,v,w)}(t_q) \end{bmatrix}, \tag{2.11}$$

where each column $\boldsymbol{U}_{H,i}^{(u,v,w)}(t_1 : t_q) \in \mathbb{R}^{m_u \times q}$ is the ith time-delayed singular mode of the Hankel matrix.

The diagonal matrix of singular values $\boldsymbol{\Sigma}_H$ is

$$\boldsymbol{\Sigma}_H = \begin{bmatrix} \sigma_{H,1} & 0 & \dots & 0 \\ 0 & \sigma_{H,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_{H,r} \end{bmatrix}, \tag{2.12}$$

and the matrix of the right singular vectors $\boldsymbol{V}_H$ can be explicitly written as

$$\boldsymbol{V}_H = \begin{bmatrix} v_{H,1}(t_1) & v_{H,1}(t_2) & \dots & v_{H,1}(t_p) \\ v_{H,2}(t_1) & v_{H,2}(t_2) & \dots & v_{H,2}(t_p) \\ \vdots & \vdots & \ddots & \vdots \\ v_{H,r}(t_1) & v_{H,r}(t_2) & \dots & v_{H,r}(t_p) \end{bmatrix}. \tag{2.13}$$

We now define the vector $\boldsymbol{v}_H(t_j) = [v_{H,1}(t_j), v_{H,2}(t_j), \dots, v_{H,r}(t_j)]^\top \in \mathbb{R}^r$ $(j = 1 \dots p)$ extracted from the columns of matrix $\boldsymbol{V}_H$. We consider $\boldsymbol{v}_H$ as the state variable of the following discrete-in-time, forced, linear dynamical system:

$$\boldsymbol{v}_H[k+1] = \boldsymbol{A}\boldsymbol{v}_H[k] + \boldsymbol{w}_2[k], \tag{2.14}$$

where $\boldsymbol{v}_H[k] = \boldsymbol{v}_H[t_k]$, $k = 1 \dots p-1$ and $\boldsymbol{A} \in \mathbb{R}^{r\times r}$.

To obtain matrix $\boldsymbol{A}$, we write (2.14) explicitly for $k = 1 \dots p-1$:

$$\boldsymbol{v}_H[2] \approx \boldsymbol{A}\boldsymbol{v}_H[1], \tag{2.15a}$$
$$\boldsymbol{v}_H[3] \approx \boldsymbol{A}\boldsymbol{v}_H[2],$$
$$\vdots \tag{2.15b}$$
$$\boldsymbol{v}_H[p] \approx \boldsymbol{A}\boldsymbol{v}_H[p-1]. \tag{2.15c}$$

In matrix form, this becomes

$$\boldsymbol{V}' \approx \boldsymbol{A}\boldsymbol{V}, \tag{2.16}$$

where $\boldsymbol{V}', \boldsymbol{V} \in \mathbb{R}^{r\times(p-1)}$:

$$\boldsymbol{V}' = \begin{bmatrix} \boldsymbol{v}_H[2] & \boldsymbol{v}_H[3] & \dots & \boldsymbol{v}_H[p] \end{bmatrix}^\top, \tag{2.17a}$$
$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_H[1] & \boldsymbol{v}_H[2] & \dots & \boldsymbol{v}_H[p-1] \end{bmatrix}^\top. \tag{2.17b}$$

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

Then the system matrix $\boldsymbol{A}$ can be calculated from

$$\boldsymbol{V}' = \boldsymbol{A}\boldsymbol{V} \Rightarrow \boldsymbol{V}'\boldsymbol{V}^\top = \boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^\top \Rightarrow \boldsymbol{A} = \boldsymbol{V}'\boldsymbol{V}^\top\left(\boldsymbol{V}\boldsymbol{V}^\top\right)^{-1}. \tag{2.18}$$

This formulation essentially applies the DMD algorithm directly to $\boldsymbol{v}_H[k]$.

Once $\boldsymbol{A}$ is known, the forcing $\boldsymbol{w}_2[k]$ can be easily computed from the training dataset, $\boldsymbol{w}_2[k] = \boldsymbol{v}_H[k+1] - \boldsymbol{A}\boldsymbol{v}_H[k]$. The covariance matrix of the forcing $\boldsymbol{Q} \in \mathbb{R}^{r\times r}$ can be obtained from

$$\boldsymbol{Q} = \mathbb{E}\left(\boldsymbol{w}_2\boldsymbol{w}_2^\top\right) = \frac{1}{p-1}\sum_{k=1}^{k=p-1} \boldsymbol{w}_2[k]\boldsymbol{w}_2^\top[k]. \tag{2.19}$$

It is important to notice that using $\boldsymbol{v}_H[k]$ we can obtain the current (at instant k) and forecast the future (at instants $k+1 \ldots k+q$) time coefficients as follows:

$$\begin{bmatrix} \boldsymbol{a}[k] \\ \boldsymbol{a}[k+1] \\ \vdots \\ \boldsymbol{a}[k+q] \end{bmatrix} = \boldsymbol{U}_H\boldsymbol{\Sigma}_H\boldsymbol{v}_H[k]. \tag{2.20}$$

In particular $\boldsymbol{a}[k]$ can be obtained from

$$\boldsymbol{a}[k] = \underbrace{\boldsymbol{U}_H(t_1)\boldsymbol{\Sigma}_H}_{\equiv \boldsymbol{C}}\boldsymbol{v}_H[k], \tag{2.21}$$

where $\boldsymbol{C} \in \mathbb{R}^{m_u\times r}$ and $\boldsymbol{U}_H(t_1) = [\ \boldsymbol{U}_{H,1}^{(u,v,w)}(t_1) \quad \boldsymbol{U}_{H,2}^{(u,v,w)}(t_1) \quad \ldots \quad \boldsymbol{U}_{H,r}^{(u,v,w)}(t_1)\ ] \in \mathbb{R}^{m_u\times r}$ is the top row of matrix $\boldsymbol{U}_H$; see (2.11).

The previous two expressions demonstrate the fundamental importance of $\boldsymbol{U}_H$. This matrix has a very clear physical interpretation, which can be understood as follows. Each column of $\boldsymbol{H}$ contains q snapshots, i.e. short time-varying sequences of fixed time duration (instead of a single snapshot as in standard POD). There are p such short sequences in total. The left singular vectors $\boldsymbol{U}_{H,i}^{(u,v,w)}$ are obtained by performing SVD on $\boldsymbol{H}$. These vectors capture the most important time-dependent patterns, in exactly the same way as the POD modes capture the most important (static) patterns from a series of snapshots. These time-varying patterns are arranged column by column in descending order of importance, as quantified by the singular values $\sigma_{H,k}$, and can be exploited to predict the future evolution from current information. They resemble Legendre polynomials for short $q \times \Delta t$ and become sinusoids for large $q \times \Delta t$ (see Dylewsky *et al.* 2022; Frame & Towne 2023). We visualise these patterns for the flow around a surface-mounted cube in §§ 4.4 and 4.5.

Note that this step is independent of the dimensionality reduction method selected in step 1. The question now is how to obtain $\boldsymbol{v}_H[k]$ since the forcing term $\boldsymbol{w}_2[k]$ in (2.14) is not known. We can get a closure if measurements at some sparse sensor points are available.

2.3. *Step 3: closure of the system using sensor measurements*

Let us assume that we have a set of l velocity measurements $\boldsymbol{s}[k] \in \mathbb{R}^l$ at a number of sensor points. At each sensor, one or more velocity components are recorded. We can express $\boldsymbol{s}[k]$ in terms of $\boldsymbol{a}[k]$ as follows:

$$\boldsymbol{s}[k] = \boldsymbol{S}\boldsymbol{U}_Y\boldsymbol{a}[k] + \boldsymbol{g}[k], \tag{2.22}$$

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

where matrix $\boldsymbol{S}$ selects the rows of the POD mode matrix $\boldsymbol{U}_Y$ corresponding to the sensor location and the velocity component being measured (it is 0 everywhere except for those points and components where the corresponding element is equal to 1). Vector $\boldsymbol{g} \in \mathbb{R}^l$ includes measurement errors as well as errors due to POD mode truncation (because only m_u modes are retained). The elements of $\boldsymbol{g}[k]$ can be obtained from the training dataset, $\boldsymbol{g}[k] = \boldsymbol{s}[k] - \boldsymbol{S}\boldsymbol{U}_Y\boldsymbol{a}[k]$), and its covariance $\boldsymbol{R} \in \mathbb{R}^{l\times l}$ can be easily calculated from

$$\boldsymbol{R} = \mathbb{E}\big(\boldsymbol{g}\boldsymbol{g}^\top\big) = \frac{1}{p-1}\sum_{k=1}^{k=p-1} \boldsymbol{g}[k]\boldsymbol{g}^\top[k]. \tag{2.23}$$

We are now ready to design a Kalman filter to estimate $\boldsymbol{v}_H[k]$ from the sparse measurements $\boldsymbol{s}[k]$. The filter takes the form

$$\hat{\boldsymbol{v}}_H[k+1] = \boldsymbol{A}\hat{\boldsymbol{v}}_H[k] + \mathcal{L}\big(\boldsymbol{s}[k] - \hat{\boldsymbol{s}}[k]\big), \tag{2.24a}$$

$$\hat{\boldsymbol{a}}[k] = \boldsymbol{C}\hat{\boldsymbol{v}}_H[k], \tag{2.24b}$$

$$\hat{\boldsymbol{s}}[k] = \boldsymbol{S}\boldsymbol{U}_Y\hat{\boldsymbol{a}}[k] = \boldsymbol{S}\boldsymbol{U}_Y\boldsymbol{C}\hat{\boldsymbol{v}}_H[k], \tag{2.24c}$$

where a hat $\hat{()}$ denotes an estimated quantity and $\mathcal{L} \in \mathbb{R}^{r\times l}$ is the Kalman filter gain. The latter is obtained from the solution of the following Riccati equation:

$$\boldsymbol{P} = \boldsymbol{A}\boldsymbol{P}\boldsymbol{A}^T - \boldsymbol{A}\boldsymbol{P}(\boldsymbol{S}\boldsymbol{U}_Y\boldsymbol{C})^T(\boldsymbol{S}\boldsymbol{U}_Y\boldsymbol{C}\boldsymbol{P}(\boldsymbol{S}\boldsymbol{U}_Y\boldsymbol{C})^T + \boldsymbol{R})^{-1}\boldsymbol{S}\boldsymbol{U}_Y\boldsymbol{C}\boldsymbol{P}\boldsymbol{A}^T + \boldsymbol{Q}, \tag{2.25a}$$

$$\mathcal{L} = \boldsymbol{A}\boldsymbol{P}(\boldsymbol{S}\boldsymbol{U}_Y\boldsymbol{C})^T(\boldsymbol{S}\boldsymbol{U}_Y\boldsymbol{C}\boldsymbol{P}(\boldsymbol{S}\boldsymbol{U}_Y\boldsymbol{C})^T + \boldsymbol{R})^{-1}. \tag{2.25b}$$

From $\hat{\boldsymbol{v}}_H[k]$ one can estimate the current and future POD coefficients from (2.20), and of course the instantaneous velocity field from (2.8).

The linear mapping between the velocity and POD coefficients, (2.22), has allowed us to synthesise a Kalman filter. This type of filter is suitable only for linear dynamical systems and linear mappings between the measurement and state variables. For nonlinear mappings, one can use an extended Kalman filter or an ensemble Kalman filter (Evensen 2003). Note also that the formulation can accommodate data from moving sensor locations (e.g. data from drones); this is achieved by using a time-dependent selection matrix $\boldsymbol{S}[k]$ in (2.22). In this case, the filter gain $\mathcal{L}[k]$ will be also time-dependent. A machine learning technique (instead of Koopman/Kalman filter theory) was proposed recently by Allen *et al.* (2025); it maps measurements $\boldsymbol{s}[k]$ directly to the future states for weather forecasting.

Steps 1 and 2 are performed offline using a training dataset. The Kalman estimator in step 3 runs online and requires only the streaming measurement data $\boldsymbol{s}[k]$. The arriving data are then mapped to the future velocity field on a rolling time window through (2.24), then (2.20) and finally (2.8). Depending on the number of retained modes m_u, it may be possible to perform such a sequential forecasting in real time, thereby making the approach also useful to experimentalists.

3. Flow forecasting from sparse scalar measurements

We follow the same approach to forecast the flow from sparse scalar measurements. First we assemble the snapshot matrix $\boldsymbol{Z}(\boldsymbol{x}, t_1 : t_K)$ of the scalar fluctuations c':

$$\boldsymbol{Z}(\boldsymbol{x}, t_1 : t_K) = \begin{bmatrix} c'(\boldsymbol{x}_1, t_1) & c'(\boldsymbol{x}_1, t_2) & \dots & c'(\boldsymbol{x}_1, t_K) \\ \vdots & \vdots & \ddots & \vdots \\ c'(\boldsymbol{x}_N, t_1) & c'(\boldsymbol{x}_N, t_2) & \dots & c'(\boldsymbol{x}_N, t_K) \end{bmatrix}, \tag{3.1}$$

where $\boldsymbol{Z} \in \mathbb{R}^{N\times K}$. Note that the scalar data are synchronised with the velocity data, that is, the time instants t_i $(i = 1 \dots K)$ in (2.1) and (3.1) are the same. As before, we apply

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

SVD to the weighted matrix $\mathcal{V}^{1/2}\boldsymbol{Z}$ (where now $\mathcal{V} = diag(V_1, V_2 \ldots V_N)$) and obtain the scalar POD modes, $U_{Z,k}(x, y, z)$, and time coefficients, $b_k(t)$. Thus we can write

$$c'(x, y, z, t) = \sum_{k=1}^{K} b_k(t) U_{Z,k}(x, y, z) \approx \sum_{k=1}^{m_c} b_k(t) U_{Z,k}(x, y, z), \tag{3.2}$$

where m_c is the number of retained scalar POD modes, or in more compact form,

$$\boldsymbol{c}'(x, y, z, t) = \boldsymbol{U}_Z(x, y, z)\boldsymbol{b}(t). \tag{3.3}$$

Equations (2.8) and (3.3) can be written together as

$$\begin{bmatrix} \boldsymbol{u}' \\ \boldsymbol{c}' \end{bmatrix}(x, y, z, t) = \underbrace{\begin{bmatrix} \boldsymbol{U}_Y & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{U}_Z \end{bmatrix}}_{\equiv \boldsymbol{U}_{YZ}} \begin{bmatrix} \boldsymbol{a}(t) \\ \boldsymbol{b}(t) \end{bmatrix}. \tag{3.4}$$

We then build the time-delayed Hankel matrix with the POD coefficients of the most dominant m_u velocity and m_c scalar modes:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{a}(t_1) & \boldsymbol{a}(t_2) & \ldots & \boldsymbol{a}(t_p) \\ \boldsymbol{b}(t_1) & \boldsymbol{b}(t_2) & \ldots & \boldsymbol{b}(t_p) \\ \boldsymbol{a}(t_2) & \boldsymbol{a}(t_3) & \ldots & \boldsymbol{a}(t_{p+1}) \\ \boldsymbol{b}(t_2) & \boldsymbol{b}(t_3) & \ldots & \boldsymbol{b}(t_{p+1}) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a}(t_q) & \boldsymbol{a}(t_{q+1}) & \ldots & \boldsymbol{a}(t_{K_{train}}) \\ \boldsymbol{b}(t_q) & \boldsymbol{b}(t_{q+1}) & \ldots & \boldsymbol{b}(t_{K_{train}}) \end{bmatrix}, \tag{3.5}$$

where $\boldsymbol{H} \in \mathbb{R}^{((m_u+m_c)\times q)\times p}$. Performing SVD on $\boldsymbol{H}$ we obtain the matrices $\boldsymbol{\Sigma}_H, \boldsymbol{V}_H, \boldsymbol{U}_H$ as before. Note that this means that we use the same weights in the velocity and scalar coefficients; this warrants further investigation, which we leave as part of future work. The matrix of the left singular vectors $\boldsymbol{U}_H \in \mathbb{R}^{((m_u+m_c)\times q)\times r}$ can be written explicitly as

$$\boldsymbol{U}_H = \begin{bmatrix} \boldsymbol{U}_{H,1}^{(u,v,w)}(t_1) & \boldsymbol{U}_{H,2}^{(u,v,w)}(t_1) & \ldots & \boldsymbol{U}_{H,r}^{(u,v,w)}(t_1) \\ \boldsymbol{U}_{H,1}^{(c)}(t_1) & \boldsymbol{U}_{H,2}^{(c)}(t_1) & \ldots & \boldsymbol{U}_{H,r}^{(c)}(t_1) \\ \boldsymbol{U}_{H,1}^{(u,v,w)}(t_2) & \boldsymbol{U}_{H,2}^{(u,v,w)}(t_2) & \ldots & \boldsymbol{U}_{H,r}^{(u,v,w)}(t_2) \\ \boldsymbol{U}_{H,1}^{(c)}(t_2) & \boldsymbol{U}_{H,2}^{(c)}(t_2) & \ldots & \boldsymbol{U}_{H,r}^{(c)}(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{U}_{H,1}^{(u,v,w)}(t_q) & \boldsymbol{U}_{H,2}^{(u,v,w)}(t_q) & \ldots & \boldsymbol{U}_{H,r}^{(u,v,w)}(t_q) \\ \boldsymbol{U}_{H,1}^{(c)}(t_q) & \boldsymbol{U}_{H,2}^{(c)}(t_q) & \ldots & \boldsymbol{U}_{H,r}^{(c)}(t_q) \end{bmatrix}. \tag{3.6}$$

A dynamical system for $\boldsymbol{v}_H$,

$$\boldsymbol{v}_H[k+1] = \boldsymbol{A}\boldsymbol{v}_H[k] + \boldsymbol{w}_2[k], \tag{3.7}$$

can be derived as before. Also, the process noise covariance $\boldsymbol{Q} \in \mathbb{R}^{r\times r}$ can be calculated in the same way as in § 2.

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

Vector $\boldsymbol{v}_H[k]$ can be used to obtain the current and future POD coefficients of the velocity and scalar fields as

$$\begin{bmatrix} \boldsymbol{a}[k] \\ \boldsymbol{b}[k] \\ \boldsymbol{a}[k+1] \\ \boldsymbol{b}[k+1] \\ \vdots \\ \boldsymbol{a}[k+q] \\ \boldsymbol{b}[k+q] \end{bmatrix} = \boldsymbol{U}_H \boldsymbol{\Sigma}_H \boldsymbol{v}_H[k]. \tag{3.8}$$

In particular, for the kth instant we have

$$\begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix}[k] = \underbrace{\boldsymbol{U}_H(t_1)\boldsymbol{\Sigma}_H}_{=\boldsymbol{C}} \boldsymbol{v}_H[k], \tag{3.9}$$

where $\boldsymbol{C} \in \mathbb{R}^{(m_u+m_c)\times r}$ and matrix $\boldsymbol{U}_H(t_1)$ represents the top two rows of $\boldsymbol{U}_H$, i.e.

$$\boldsymbol{U}_H(t_1) = \begin{bmatrix} \boldsymbol{U}_{H,1}^{(u,v,w)}(t_1) & \boldsymbol{U}_{H,2}^{(u,v,w)}(t_1) & \dots & \boldsymbol{U}_{H,r}^{(u,v,w)}(t_1) \\ \boldsymbol{U}_{H,1}^{(c)}(t_1) & \boldsymbol{U}_{H,2}^{(c)}(t_1) & \dots & \boldsymbol{U}_{H,r}^{(c)}(t_1) \end{bmatrix}. \tag{3.10}$$

Let us assume that we have now l scalar measurements $\boldsymbol{s}[k]$; they can be written as

$$\boldsymbol{s}[k] = \boldsymbol{S}\boldsymbol{U}_Z \boldsymbol{b}[k] + \boldsymbol{g}[k], \tag{3.11}$$

where now matrix $\boldsymbol{S}$ selects the rows of the POD mode matrix $\boldsymbol{U}_Z$ corresponding to the scalar sensor locations. Note that it is possible to mix velocity and scalar measurements; in this case $\boldsymbol{S}$ will act on the compound matrix $\boldsymbol{U}_{YZ}$ (see (3.4)). In the following we assume that we have scalar measurements only. The covariance $\boldsymbol{R}$ of vector $\boldsymbol{g}$ can be obtained as explained in the previous section.

The Kalman filter takes the form

$$\hat{\boldsymbol{v}}_H[k+1] = \boldsymbol{A}\hat{\boldsymbol{v}}_H[k] + \mathcal{L}\big(\boldsymbol{s}[k] - \hat{\boldsymbol{s}}[k]\big), \tag{3.12a}$$

$$\begin{bmatrix} \hat{\boldsymbol{a}} \\ \hat{\boldsymbol{b}} \end{bmatrix}[k] = \boldsymbol{C}\hat{\boldsymbol{v}}_H[k], \tag{3.12b}$$

$$\hat{\boldsymbol{s}}[k] = \boldsymbol{S}\boldsymbol{U}_Z\hat{\boldsymbol{b}}[k] = \boldsymbol{S}\boldsymbol{U}_Z(\boldsymbol{C})_2\hat{\boldsymbol{v}}_H[k], \tag{3.12c}$$

where $(\boldsymbol{C})_2$ indicates the second-row block of matrix $\boldsymbol{C}$, and the Kalman gain matrix $\mathcal{L}$ is obtained by solving a Riccati equation similar to (2.25) where $\boldsymbol{U}_Y$ is replaced by $\boldsymbol{U}_Z$.

From $\hat{\boldsymbol{v}}_H[k]$ we can estimate the current and future POD coefficients from (3.8) and the instantaneous velocity and scalar fields from (3.4).

4. Application to the flow around a surface-mounted cube

4.1. *Computational set-up and numerical methodology*

We consider the three-dimensional flow around a surface-mounted cube of height h. The computational domain, shown in figure 1, has dimensions $L_x \times L_y \times L_z = 19h \times 10h \times 10h$. The origin of the coordinate system is located at the bottom midpoint of the upstream face of the cube. The inlet is located at $x/h = -6$ in the streamwise direction and the outlet at $x/h = 13$. The domain extends between $-5 \leqslant z/h \leqslant 5$ in the spanwise direction and up

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

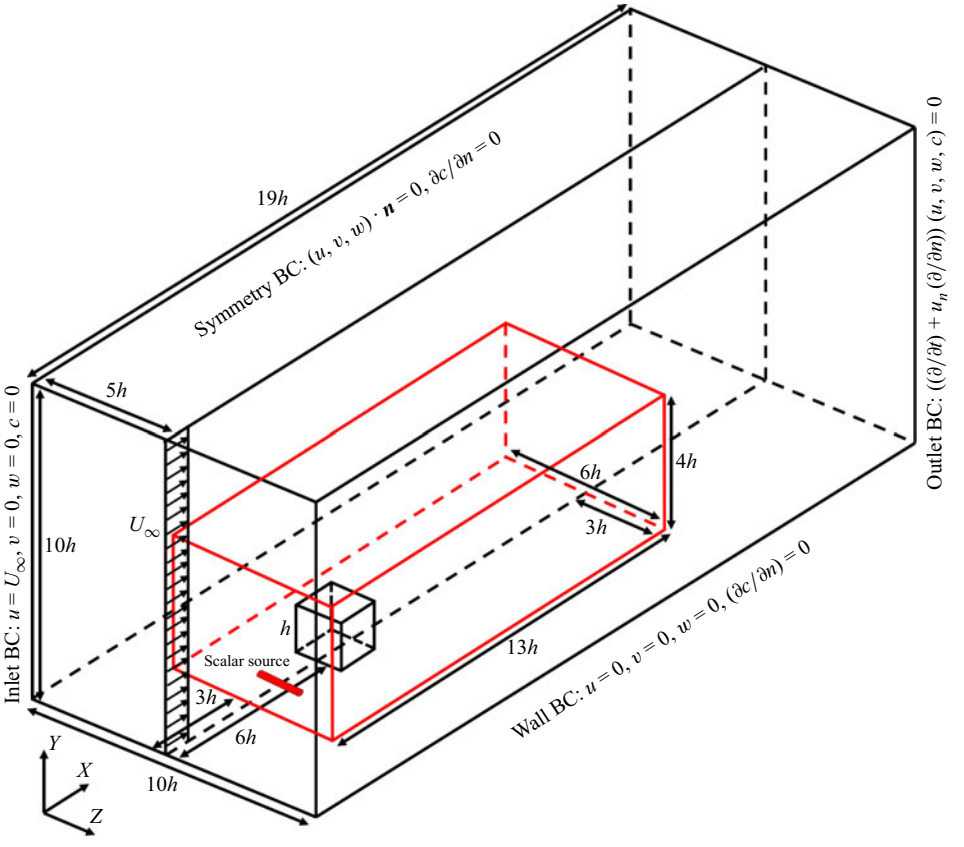


Figure 1. Computational domain and boundary conditions (BC). The red lines mark the boundaries of the region where snapshot data are collected.

to $y/h = 10$ in the wall-normal direction. Uniform velocity $U_\infty = 1$ is prescribed at the inlet and a convective boundary condition at the outlet. No-slip conditions are imposed on the cube surfaces and bottom wall, while symmetry conditions are applied on the top and spanwise boundaries (Krajnovic & Davidson 2002).

A scalar is released from a source with elliptical cross-section centred at $(x_s, y_s, z_s) = (-2, 0.1, -0.5 : +0.5)h$. The axis of the source is along the z direction and at the same elevation as the core of the horseshoe vortex forming in front of the cube (see later). The source strength $\hat{m}_T(\boldsymbol{x})$ (amount of scalar released per unit volume) is steady with a spatial distribution given by

$$\hat{m}_T = 2\gamma\,[\cos(r\pi) + 1]\,, \tag{4.1}$$

with

$$\gamma = \frac{1}{4r_x r_y(\pi - 2/\pi)}, \quad r = \sqrt{\left(\frac{x - x_s}{r_x}\right)^2 + \left(\frac{y - y_s}{r_y}\right)^2}, \tag{4.2}$$

where (x_s, y_s, z_s) are the coordinates of the centre of the source, $r_x = 0.1h$ is the major axis and $r_y = 0.08h$ is the minor axis (smaller than r_x due to the presence of the bottom wall). Equation (4.1) represents a source distribution that vanishes at the boundary of the elliptical cross-section. The normalisation parameter γ ensures the volume integral of $\hat{m}_T$ is unity.

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

The flow is simulated with the in-house code PANTARHEI. The incompressible Navier–Stokes equations are discretised in space using the finite-volume method, with a second-order central discretisation scheme (for both convection and viscous terms) and marched in time with a third-order backward scheme. The fractional step method is employed to obtain pressure and correct the velocities to satisfy the continuity equation. The code has been used extensively to simulate transitional and turbulent flows (Thomareis & Papadakis 2017, 2018; Yao & Papadakis 2023; Schlander, Rigopoulos & Papadakis 2024).

The Reynolds number, defined as $Re_h = U_\infty h/\nu$, is set to 5000. The flow domain is discretised using $N_x \times N_y \times N_z = 576 \times 268 \times 406$ cells in a Cartesian arrangement; the total number around the cube is 61 612 200. The cells are clustered close to the cube surfaces and bottom wall. Each cube edge is discretised with 102 cells. The thickness is $0.006h$ in the first layer near the walls. A time step $\delta t U_\infty / h = 0.001$ is selected to satisfy CFL < 0.5. The maximum ratio of the cell size (taken as the cubic root of cell volume) to the Kolmogorov length scale was equal to 4.6 in the recirculation zone behind the cube, which indicates almost direct numerical simulation (DNS)-quality resolution, which is sufficient for the purposes of the present investigation.

Snapshots of velocities and scalar fields are collected within a subregion (defined by the red lines of figure 1) that contains $N = 38.69 \times 10^6$ cells. The flow is first advanced for 2 flow-through times; the simulation is then restarted and advanced for 6 more flow-through times. In total $K = 6000$ flow and scalar field snapshots are recorded synchronously during the last 6 flow-through times. The time separation between snapshots is $\Delta t = 0.019 h/U_\infty$.

4.2. *Time-averaged flow and scalar fields*

Figure 2 shows time-averaged streamlines superimposed on the mean pressure fields on the symmetry xy plane and the xz plane at the height of the first cell centroid away from the bottom wall. The reference pressure is the mean static pressure at the inlet plane. In figure 2(a), three main separation regions can be seen, at the front (F), at the leading edge (T) and downstream of the cube (R). There is a secondary recirculation region (N) at the bottom corner of the leeward face.

In figure 2(b), it can be seen that the flow separates upstream of the cube, at the saddle point S_s located at $(-1.8, 0, 0)h$. The flow reattaches upstream of the cube at the nodal point N_a at $(-0.015, 0, 0)h$. The streamline passing through S_s bends around the cube and forms a 'line of separation' (LS) close to the bottom wall. The 'line of attachment' (LA) is formed by streamlines passing through N_a. The horseshoe vortex centre forms a line between LS and LA and is deflected around the cube to form two legs B and B'. The shear layers separating from the two vertical edges of the front face generate the two lateral vortices E and E'. The two vortices and the shear layer over the cube join at a higher elevation to form an arc-shaped vortex tube. Downstream of the cube, two symmetrically located points D_1 and D_2 indicate two vortices on the bottom wall. Further downstream, the flow reattaches at S_r $(3.31, 0, 0)h$. The reattachment length is $L_R/h = 2.31$.

Contour plots of the Reynolds stresses and the turbulent kinetic energy (TKE) at the symmetry plane $z/h = 0$ are shown in figure 3. High levels of $\langle u'u' \rangle$ are found inside the separating shear layer emanating from the leading edge. The peak value is located above the cube at $(x, y) = (0.9, 1.36)h$, while $\langle v'v' \rangle$ peaks downstream of the cube at $(x, y) = (2.85, 0.71)h$. The peak of $\langle u'v' \rangle$ is located at $(x, y) = (0.95, 1.36)h$ which is close to the peak of $\langle u'u' \rangle$. Finally, the TKE peak is located at $(x, y) = (2.6, 0.86)h$ closer to the peak location of $\langle v'v' \rangle$. Small patches of Reynolds stresses can be seen upstream of the cube, around the horseshoe vortex centre.

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

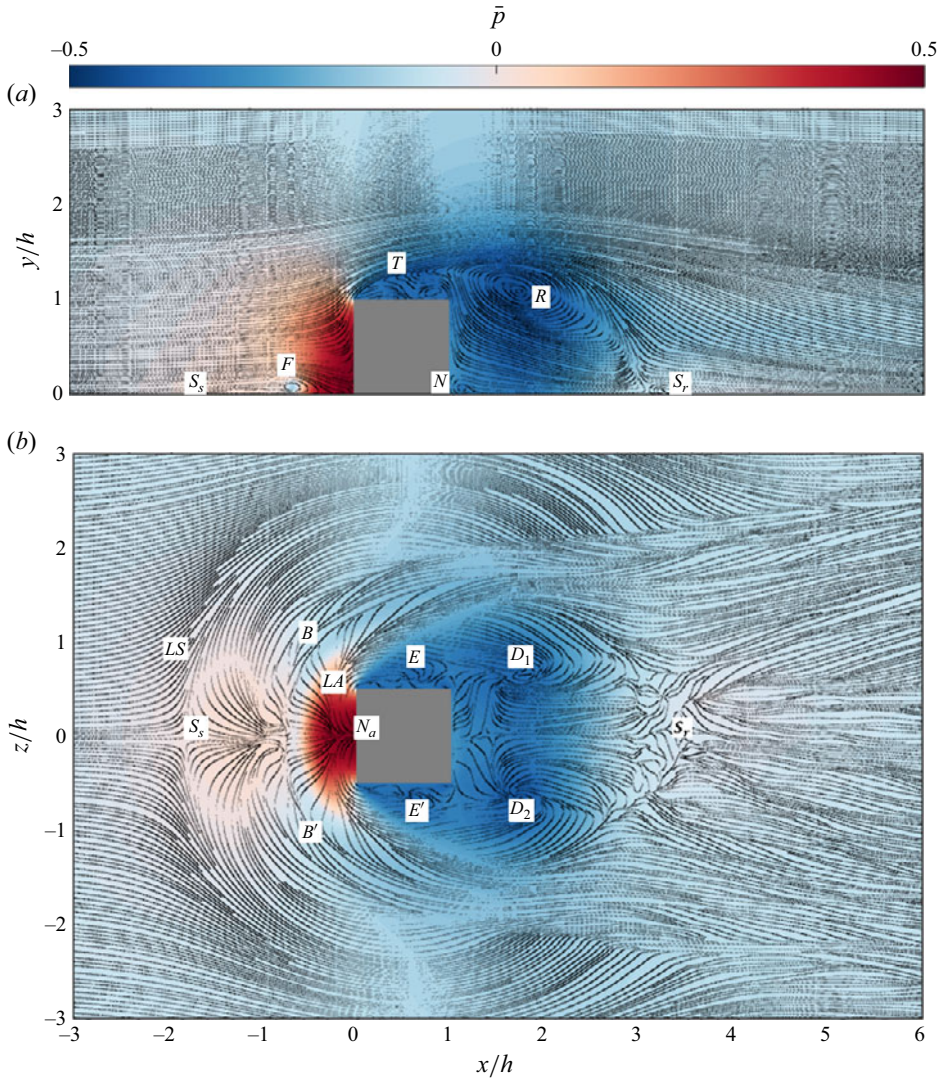


Figure 2. Time-averaged streamlines superimposed on contours of mean pressure field: (*a*) symmetry xy plane at $z/h = 0$; (*b*) xz plane at distance $y/h = 0.003$ from the bottom wall.

Figure 4 shows contours of Reynolds stresses and TKE in the xz plane at mid-height $y/h = 0.5$. High levels of $\langle u'u' \rangle$ and $\langle u'w' \rangle$ are found inside the shear layers separating from the front vertical edges and downstream of the cube. The peak value of $\langle u'u' \rangle$ is found at $(x, z) = (0.8, \pm 0.85)h$. Notice the high values of the normal stresses in the spanwise direction $\langle w'w' \rangle$ that peak further downstream at $(x, z) = (3.15, 0)h$. Such high values indicate strong symmetry-breaking motions. The TKE combines features of $\langle u'u' \rangle$ and $\langle w'w' \rangle$ but is more influenced by the spanwise stresses.

Mean and root-mean-square (r.m.s.) profiles of the streamwise velocity are compared against the experimental data of Castro & Robins (1977) in figure 5. At $x/h = 0.5$ and 1.5 the DNS solution follows the experimental results quite well. At $x/h = 2.5$, the numerical results are also good, but slightly overestimate the backflow streamwise velocity within

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

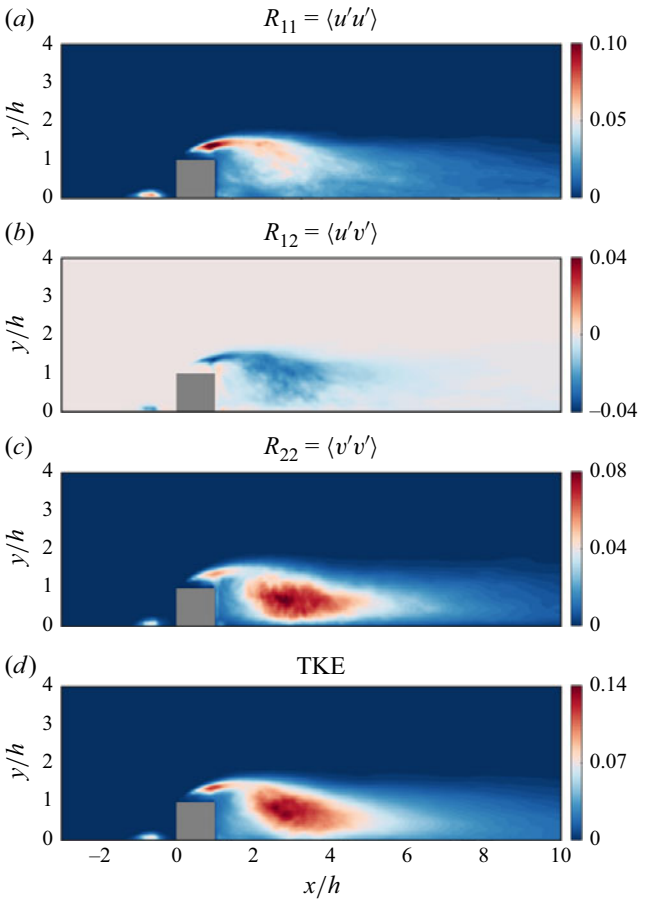


Figure 3. Contours of the (*a–c*) Reynolds stresses and (*d*) TKE at the symmetry xy plane at $z/h = 0$.

the recirculation zone in $y/h = 0{-}1$. For the r.m.s. of streamwise velocity, the numerical results give quantitatively good predictions at all streamwise positions. Small differences can be attributed to different inlet velocity profiles. We have used a constant profile at the inlet, but for the experiments a fully developed boundary-layer profile was employed. We also compare our predictions with the DNS of Rossi, Philips & Iaccarino (2010). Figure 6 presents profiles of normal stresses $\langle v'v' \rangle$, $\langle u'v' \rangle$. The present results compare well with those of the literature at all locations.

Contours of the Kolmogorov time scale $\eta_t = \sqrt{\nu/\langle \epsilon \rangle + 0.001}$, where ν is the kinematic viscosity and $\langle \epsilon \rangle$ the mean dissipation rate, are shown in figure 7 at two planes. The areas with meaningful values are in the separating shear layers and inside the recirculation zone. The η_t values range between 0.03 and 0.2 time units (one time unit is equal to h/U_∞). It is expected that the time scale associated with the maximum Lyapunov exponent will be of similar magnitude, as explained in the Introduction.

Contours of the mean and r.m.s. of scalar on the xz plane at $y/h = 0.1$ are shown in figure 8. The scalar is convected towards the cube, bends around it and due to dilution effect the concentration drops. The scalar dispersion is dominated by the horseshoe vortex near the cube. The scalar r.m.s. values are maximised in the region where the flow around

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

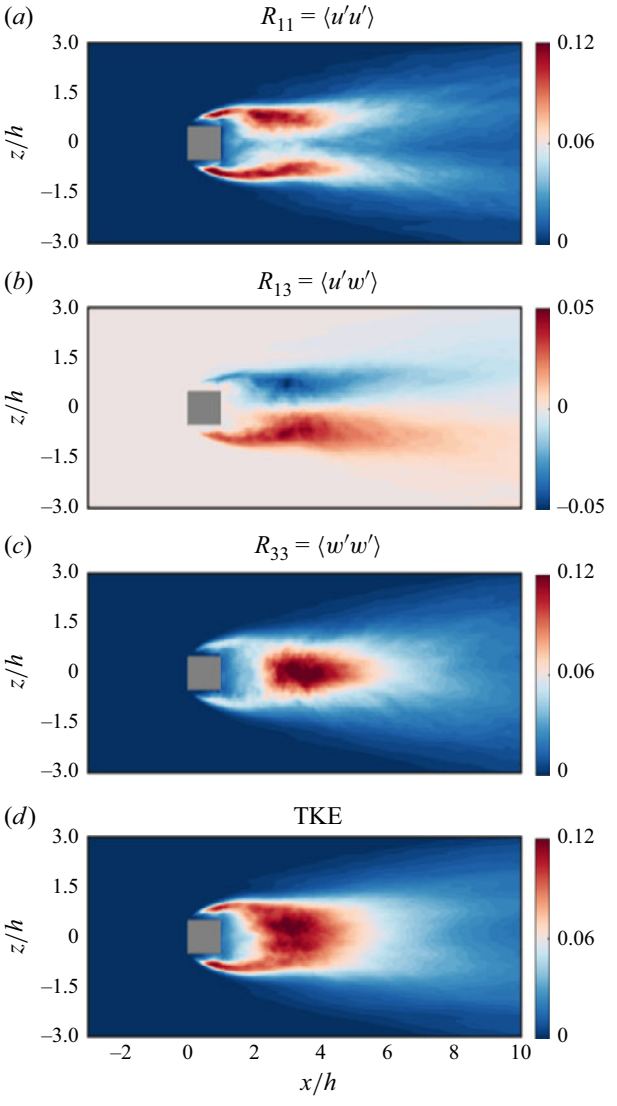


Figure 4. Contours of the (*a–c*) Reynolds stresses and (*d*) TKE in the xz plane at mid-height $y/h = 0.5$.

the cube bends. This is the area of the highest mean scalar gradient (see figure 8*a*) and turbulent generation due to the separating shear layers from the vertical edges of the front face.

4.3. *Proper orthogonal decomposition modes*

We implemented the sequential SVD method of Li, Hulshoff & Hickel (2021) to obtain the POD eigenvectors and time coefficients from $K = 6000$ velocity and scalar snapshots. We did convergence tests and for all the m_u (retained velocity POD modes) considered we get eigenvalue convergence even with 4000 snapshots. The sequential method reduces significantly the memory requirements. Figure 9(*a*) shows the energy fraction of each mode in log scale. It can be seen that the first two modes have the same energy (about 10 % of the total) and they are paired (as will be seen later). Modes 3–8 have similar

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

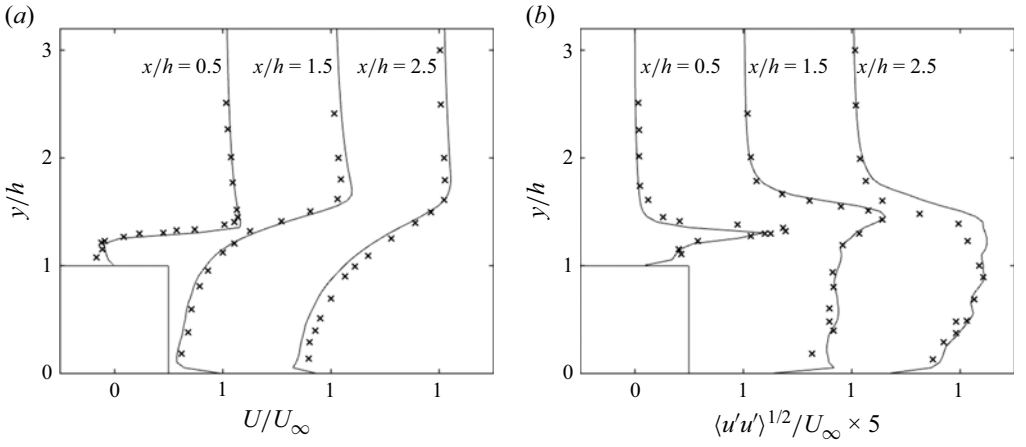


Figure 5. Comparison of numerical results with measurements of (*a*) mean and (*b*) r.m.s. of streamwise velocity in the symmetry xy plane at $z/h = 0$: −, present DNS; ×, Castro & Robins (1977).

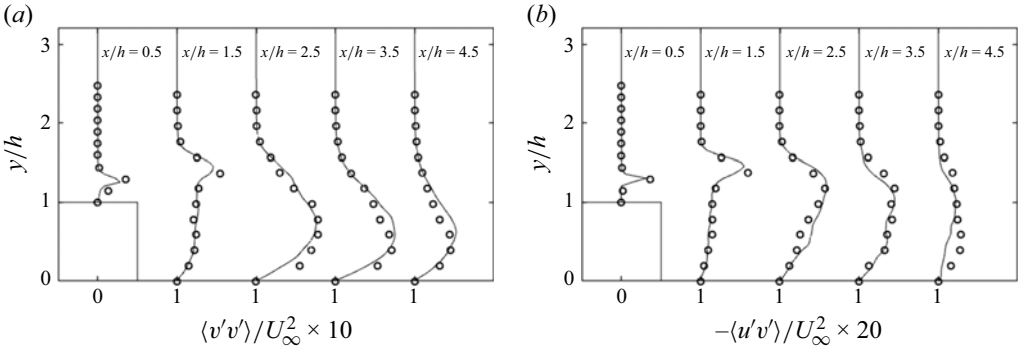


Figure 6. (*a*,*b*) Comparison of present results (solid lines) for Reynolds stresses at the symmetry xy plane at $z/h = 0$ against the DNS results (circles) of Rossi *et al.* (2010).

energy (1−3 %), all other modes having energy less than 1 %. It is very interesting to note that the energy distribution of the higher modes (with index larger than 100) follows a power law with slope $-11/9$. This power law was first derived theoretically in Knight & Sirovich (1990) from the well-known $\kappa^{-5/3}$ law for the energy density with respect to wavenumber κ in the inertial regime for homogeneous isotropic turbulence. The fact that the energy follows the $-11/9$ power law confirms that the turbulent flow is well resolved. Figure 9(*b*) displays the cumulative energy, the first 523 modes (shaded area) accounting for 97 % of the total energy.

Iso-surfaces of the three dominant modes are shown in figure 10. The streamwise modes $U^{(u)}_{Y,1}$ and $U^{(u)}_{Y,2}$ illustrate three-dimensional structures that are antisymmetric with respect to the xy $z/h = 0$ plane and are shed downstream in alternating fashion. Similar antisymmetric and alternating behaviour can also be observed for the wall-normal modes $U^{(v)}_{Y,1}$ and $U^{(v)}_{Y,2}$. The spanwise velocity modes ($U^{(w)}_{Y,1}$ and $U^{(w)}_{Y,2}$) on the other hand consist of distinct structures which are symmetric and are spatially shifted in the streamwise direction. This spatial shift combined with the temporal shift between the POD coefficients (not shown) and the sharp spectral peak at the same frequency as is shown

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

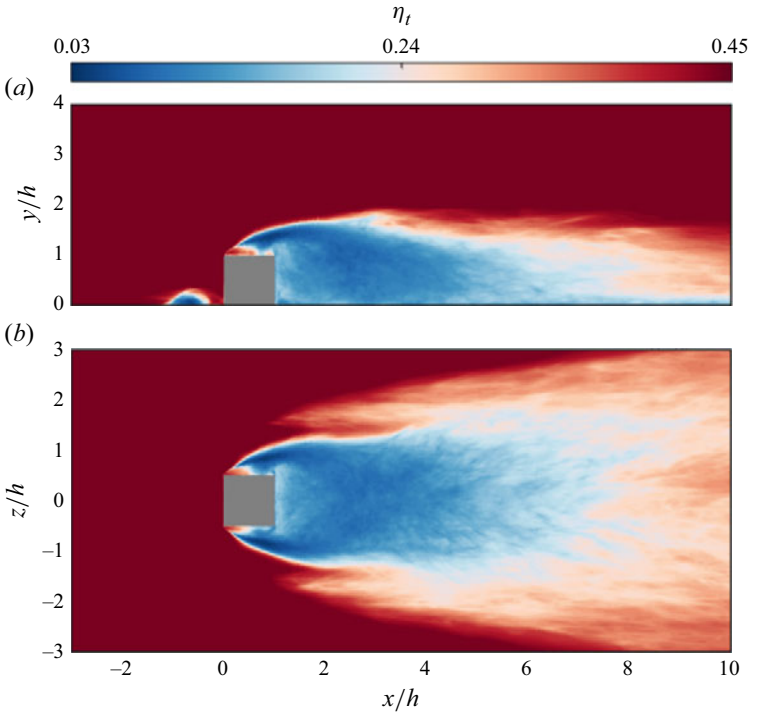


Figure 7. Contour plots of the Kolmogorov time scale η_t: (*a*) symmetry xy plane at $z/h = 0$; (*b*) xz plane at mid-height $y/h = 0.5$.

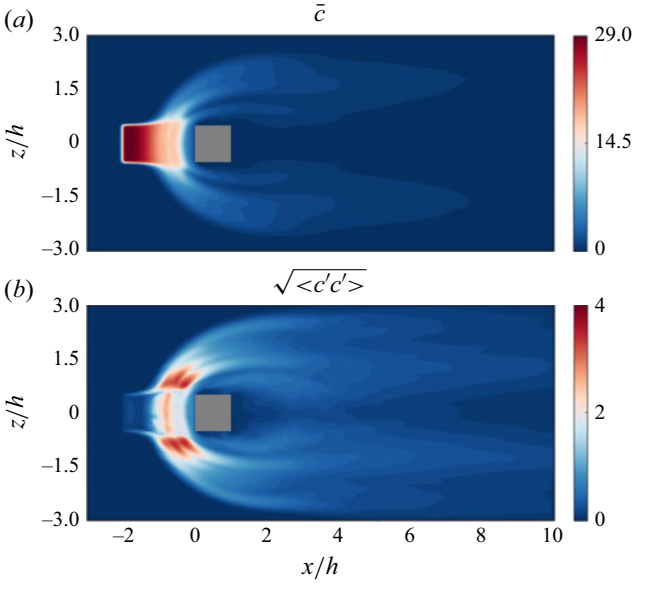


Figure 8. Contour plots of the (*a*) mean scalar field and (*b*) r.m.s. of scalar field in the xz plane at $y/h = 0.1$ (height of the source centre).

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

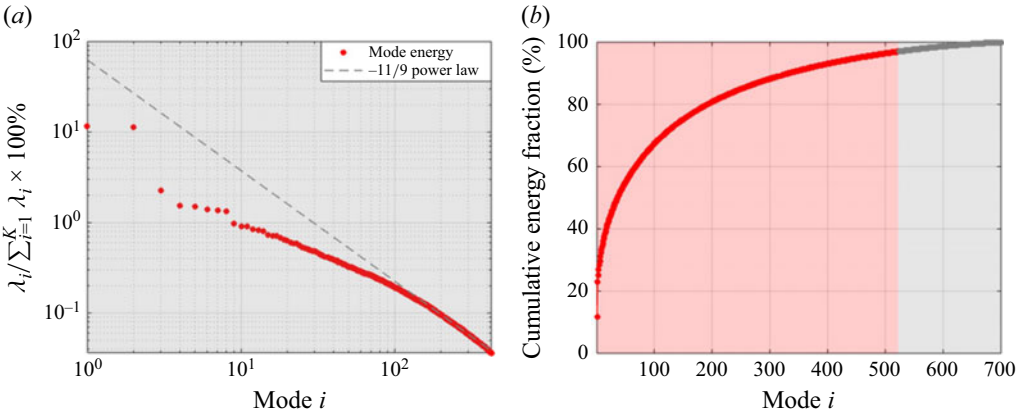


Figure 9. Energy fraction of (*a*) the first 424 velocity POD modes and (*b*) the cumulative energy.

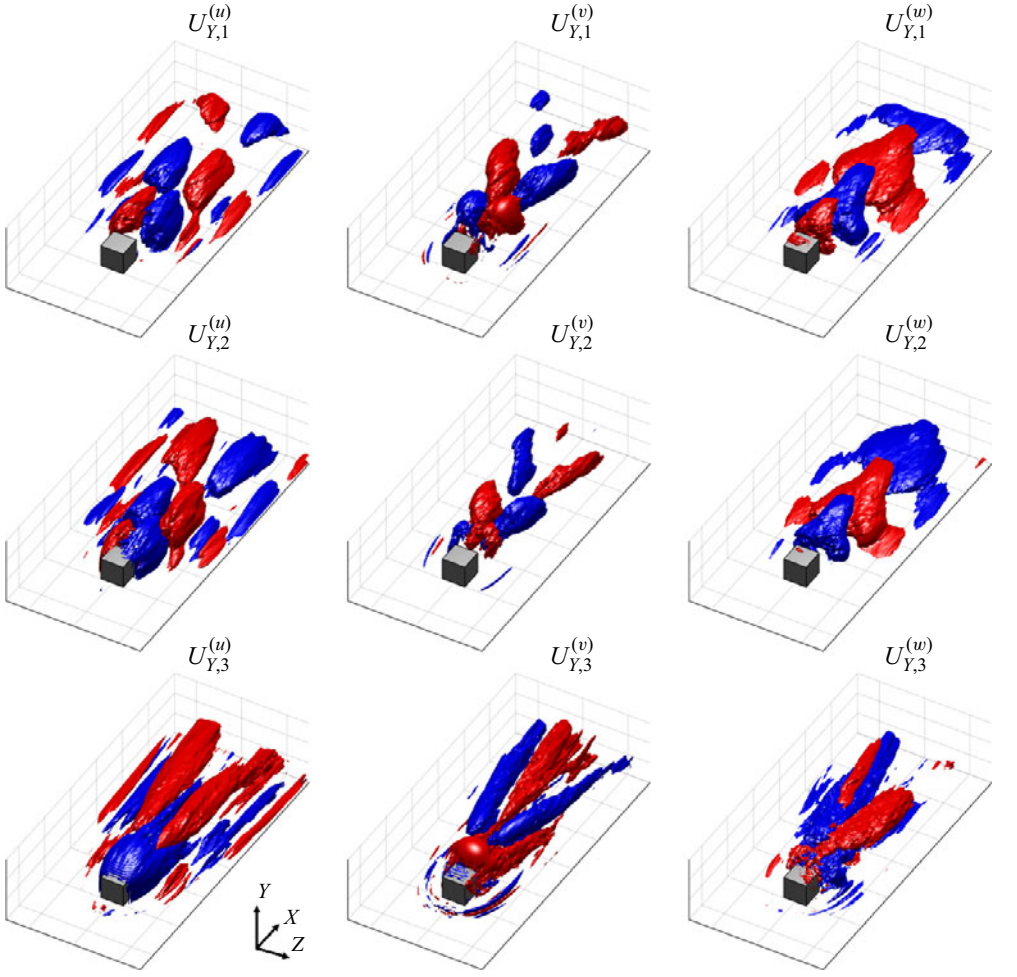


Figure 10. Iso-surfaces of (*a*) $U_{Y,i}^{(u)}$, (*b*) $U_{Y,i}^{(v)}$ and (*c*) $U_{Y,i}^{(w)}$ for POD modes 1−3. Iso-surfaces of the velocity modes are normalised with the L_∞-norm: $U_{:,1}$ and $U_{:,2}$ blue (−0.2), red (+0.2); $U_{:,3}$ blue (−0.16), red (+0.16).

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

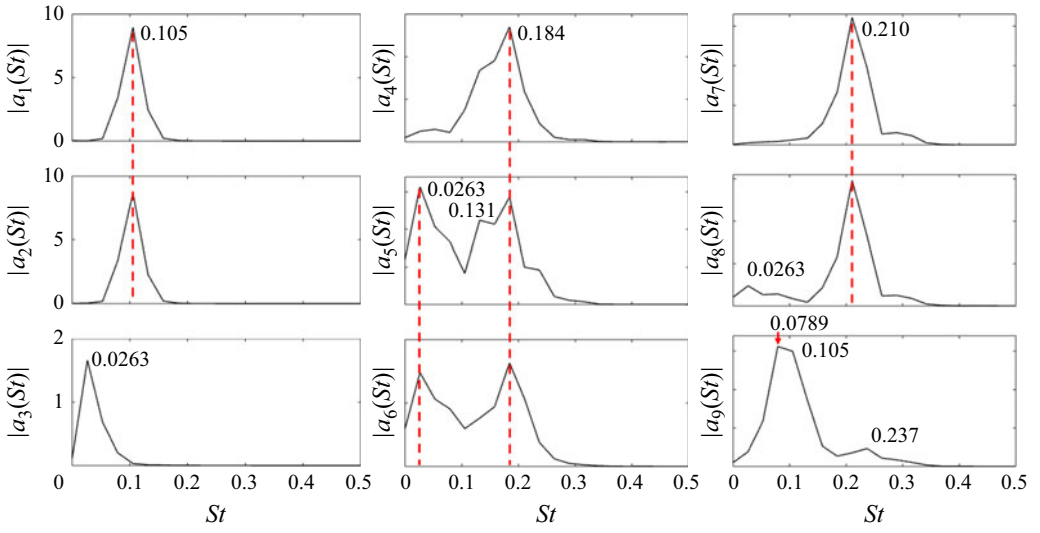


Figure 11. Spectra of time coefficients $a_i(t)$ of the nine most dominant POD modes.

later results in downstream-propagating structures. The eigenvectors of the streamwise and wall-normal velocities of the third mode are symmetric and consist of shear-layer structures originating from the upstream edges of the cube. Moreover, a single pair of counter-rotating streamwise structures is also observed in iso-surfaces of $U^{(u)}_{Y,3}$ and $U^{(v)}_{Y,3}$ downstream of the cube (see also Bourgeois, Sattari & Martinuzzi 2011). Higher-order modes consist of smaller structures that are difficult to visualise and interpret.

The spectra of the POD coefficients of the nine most dominant modes are shown in figure 11. The spectra were obtained using the Hanning windowing method. The time signal was split in five segments with an overlapping ratio of 50 %. The first two modes peak at the same frequency $St \approx 0.105$, where $St = fh/U_\infty$. A low-frequency peak $St = 0.0263$ with high intensity is found in the spectrum of the third mode. This low-frequency peak persists in the spectra of modes 5 and 6. Mode 4 peaks at $St = 0.184$ ($\approx 2 \times 0.105 - 0.0263$, i.e. a linear combination of the previous two frequencies). The first harmonic of modes 1 and 2 appears in the spectra of modes 7 and 8.

Figure 12 shows the variance fraction of the scalar POD modes. It also follows the $-11/9$ power law for mode numbers larger than $\approx$70. The first 330 modes (shown in red-shaded area) account for 97 % of scalar variance. Thus fewer modes are needed for the scalar compared with the flow field to capture the same percentage of the variance.

Sensors are placed at the peaks of the velocity POD modes. The sensors record the three velocity fluctuations $u'(t)$, $v'(t)$ and $w'(t)$. The sensor locations from the leading 10 modes superimposed on contours of the mean vorticity and TKE are shown in figure 13. As intuitively expected, the sensors are clustered in the region of high TKE. Scalar sensors are also placed at the same locations. The distribution of the sensor locations is close to, but not perfectly, symmetric about the xy plane at $z/h = 0$ in the wake. As can be seen from figure 10, the modes are symmetric or antisymmetric with respect to the symmetry plane. The sensors are placed at the peak location of the POD mode magnitude, but only one peak is randomly selected, which can be on either side of the symmetry plane. Since only 10 sensor locations are shown, this does not average out to a fully symmetric picture. In practice, we also expect very small discrepancies from perfect symmetry in the computed

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

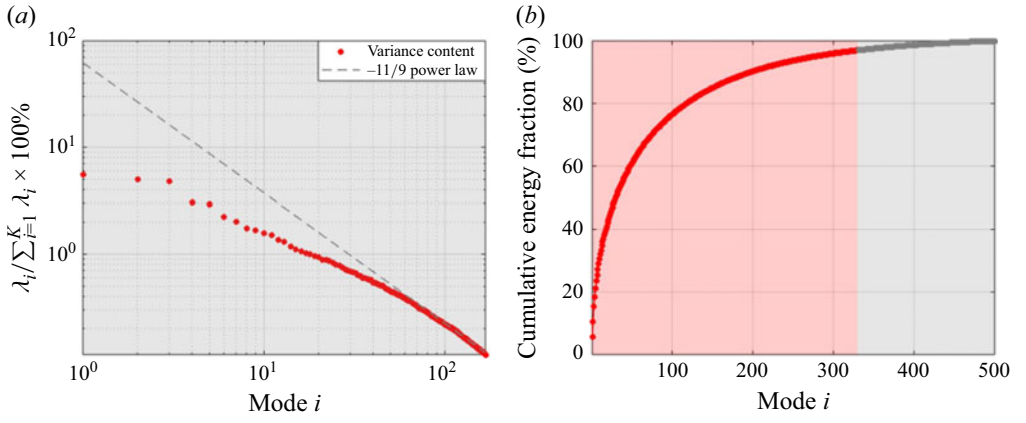


Figure 12. Variance fraction of (*a*) the first 167 scalar POD modes and (*b*) the cumulative variance: snapshots number $K = 6000$.

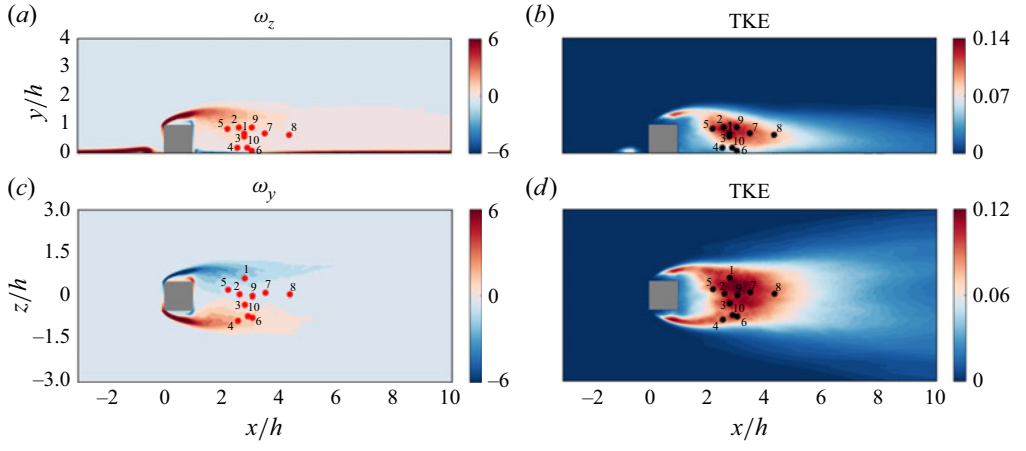


Figure 13. Locations of the first 10 sensors placed at the peaks of the 10 most dominant velocity POD modes superimposed on contours of spanwise (*a*) and wall-normal vorticity (*c*) and TKE at the symmetry xy plane (*b*) and xz plane at $y/h = 0.5$ (*d*).

modes of a fully turbulent three-dimensional inhomogeneous flow, especially for high-order modes that contain very little energy, of the order of 1 % or less as seen in figure 9.

4.4. *Flow-field reconstruction and forecasting from velocity measurements*

We first use $q = 1$ and assemble the Hankel matrix using time coefficients of the first m_u velocity POD modes. The parameter m_u is varied from 18, 37 to 68 accounting for 40 %, 50 % and 60 % of the TKE content, respectively. We use $K_{train} = 4000$ snapshots to extract the model matrices (training dataset) and the rest of the snapshots (i.e. 2000) to validate the estimator (validation dataset). The reconstruction quality is quantified with the FIT[%] metric which is defined as

$$\mathrm{FIT}[\%] = 100\left(1 - \frac{\sum_{i=1}^{10}\overline{\left(a_i[k] - \hat{a}_i[k]\right)^2}}{\sum_{i=1}^{10}\overline{a_i^2[k]}}\right), \tag{4.3}$$

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

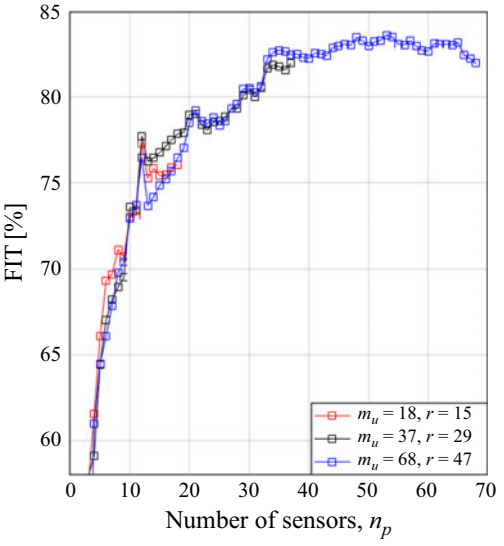


Figure 14. The FIT[%] metric against the number of velocity sensors, n_p.

where the overbar denotes average over the index k in the validation dateset. This metric quantifies the difference between the true and estimated time coefficients for the first 10 POD modes (that contain 35 % of the total energy). The flow is turbulent and hundreds of POD modes are needed to capture the kinetic energy of the system, as shown in figure 9(*b*). Higher-order modes have narrow spatial footprint; it is thus unrealistic to expect that a small number of sensors will be able to reconstruct the total energy. For this reason, we target the energy of the 10 most dominant modes.

In figure 14, the FIT[%] metric for different m_u velocity modes and number of sensors n_p is shown. The parameter r is the model order and is determined so as to capture 97 % of the SVD content of the Hankel matrix. It can be seen that the reconstruction performance is practically insensitive to m_u; the three curves almost collapse. The number of sensors n_p is limited by the order r of the model; larger m_u increases the model order and enables the use of more sensors. Note the rapid growth of the FIT[%] metric with n_p; the reconstruction achieves high accuracy with only approximately 15 sensors. We expect this number to be flow-dependent. For larger n_p, the performance plateaus.

The reconstruction quality can be further assessed by comparing the Reynolds stress fields obtained from the true and estimated time coefficients. We select $m_u = 68$, $n_p = 53$ that gives the best reconstruction performance. In figure 15, we compare the Reynolds stress and TKE fields evaluated using the true (figure 15*a*,*c*,*e*,*g*) and estimated (figure 15*b*,*d*,*f*,*h*) time coefficients of the first 10 modes. The same colour scale has been employed to facilitate the comparison. It can be seen that figure 15(*b*,*d*,*f*,*h*) reproduces satisfactorily the true flow statistics. The reconstructed normal stresses $\langle u'u' \rangle$ and $\langle w'w' \rangle$ have a more confined region of high values compared with the true statistics and this affects also the TKE. The $\langle u'w' \rangle$ reconstruction is satisfactory.

To explain the observed behaviour, the metric $\mathrm{FIT}_i[\%]$ for each individual mode, defined as

$$\mathrm{FIT}_i[\%] = 100\left(1 - \frac{\left\|a_i[k] - \hat{a}_i[k]\right\|}{\left\|a_i[k] - \overline{a_i[k]}\right\|}\right), \tag{4.4}$$

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

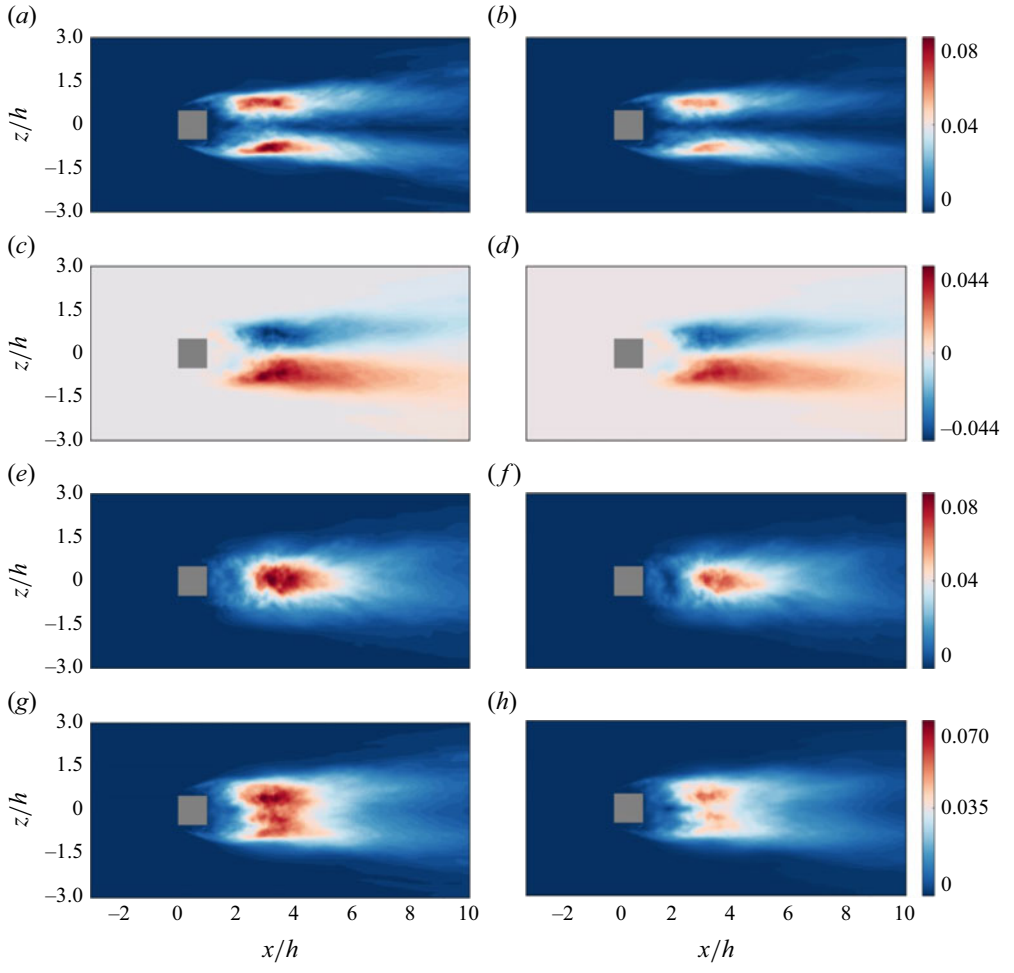


Figure 15. Flow statistics: (a,b) $\langle u'^2\rangle$, (c,d) $\langle u'w'\rangle$, (e,f) $\langle w'^2\rangle$ and (g,h) TKE. Statistics obtained from the true time coefficients a_i (a,c,e,g) and the estimated time coefficients $\hat{a}_i$ (b,d,f,h) of the first 10 POD modes.

was evaluated for the training and validation datasets; the results are shown in figure 16. The reconstruction quality in the training dataset is remarkably good for all the modes considered here. It can be seen that the first two modes are very well reconstructed in the validation dataset, with $\mathrm{FIT}_i[\%]$ being more than 80 %. Modes 3−8 have lower reconstruction quality, about 35−40 %, and modes 9 and 10 less than 20 %. The first two modes account for 23 % of the total energy and make up 67 % of the energy for the first 10 modes. The modest reconstruction quality for higher modes is responsible for the discrepancies in the flow statistics shown in figure 15.

To assess the quality of forecasting of the future evolution of the flow field, we now explore different values of $q = 120$, 239, 478, 717 and 956 that correspond to time-delay windows $q \times \Delta t = 2.28$, 4.54, 9.08, 13.62 and 18.16, respectively. Note that the time delay $q \times \Delta t = 9.08$ is close to the period of the first two POD modes, and is more than one order of magnitude larger than the Lyapunov time scale (see figure 7). The largest forecasting window considered, $q \times \Delta t = 18.16$, is two orders of magnitude larger. We used a training dataset of $K_{train} = 5000$ snapshots (that correspond to 95 time units) to construct the linear model and the estimator as explained in § 2. We used more training

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

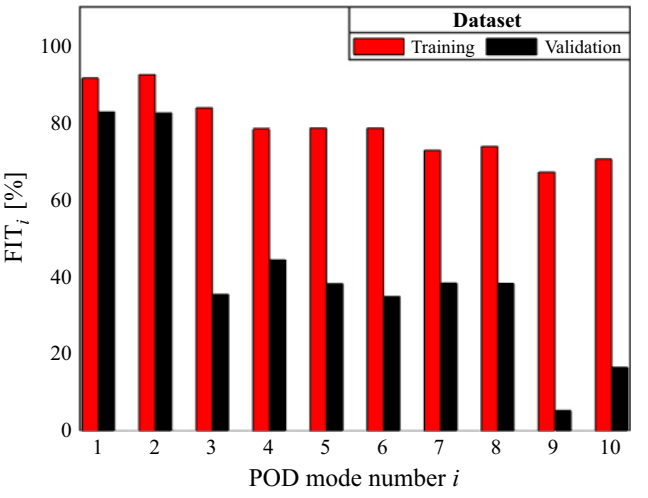


Figure 16. The FIT$_i$[%] metric of the first 10 modes for $m_u = 68$, $r = 38$, $n_p = 53$. Red: training dataset. Black: validation dataset.

data (5000 instead of 4000 snapshots when $q = 1$) because forecasting is more challenging and also we do not need to get statistics (i.e. we do not need a long validation dataset).

We first visualise the left singular vectors $\boldsymbol{U}_{H,i}^{(u,v,w)}$ that form the columns of matrix $\boldsymbol{U}_H$; see (2.11). Contour plots of the first six of those vectors in the time-delay/mode-number plane are shown in figure 17 for $m_u = 18$ retained velocity POD modes, $q = 956$, $q \times \Delta t = 18.16$ (approximately equal to two shedding periods). It can be seen that the first two singular vectors encode the time-periodic behaviour of modes 1 and 2; indeed the footprint is strongest for $k = 1$ and $k = 2$. Note also that the two subplots are shifted in the time-delay axis. In the third singular vector the footprint is strong for $k = 3$ over a long time delay, reflecting the features of the slow third POD mode. Higher-order singular vectors have more complicated patterns over all values of k and are more difficult to interpret. For the three-equation Lorenz system, Brunton *et al.* (2017) found that the shape of the singular vectors take the form of polynomials. This was because the time delay was short. In the present case, where the time delay is longer, the modes become periodic, as expected from theoretical analysis (see Frame & Towne 2023). The message from this plot is clear; the time history of the flow has left its imprint in these vectors and it is exactly this information that allows the forecasting of the future evolution of the flow from current conditions.

We now examine how the FIT[%] metric changes with the number of POD modes m_u and the number of sensors n_p; results are shown in figure 18 for the three largest values of q examined. It can be seen that the FIT[%] curves almost collapse and the forecasting quality is almost independent of m_u. Only for the largest $q \times \Delta t$ is there some small deviation between the three curves. Note that again, as with $q = 1$, FIT[%] reaches a plateau with relatively few sensors. Most importantly, as q increases the forecasting quality decreases but only slightly; this indicates that it is robust to the size of the foresting window.

There is another important observation that we can make by inspecting figure 18. The performance of reconstruction from a small number of sensors (even one sensor) improves significantly as the time delay $q \times \Delta t$ grows; compare for example FIT[%] across the three values of $q \times \Delta t$ when $n_p = 1$ sensor is used. Especially for the largest time delay, adding more sensors only marginally increases the FIT[%] values. This trend agrees with the findings of Takens (1981) who recognised the usefulness of time-delayed embeddings of even a single observable for the analysis of chaotic systems.

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

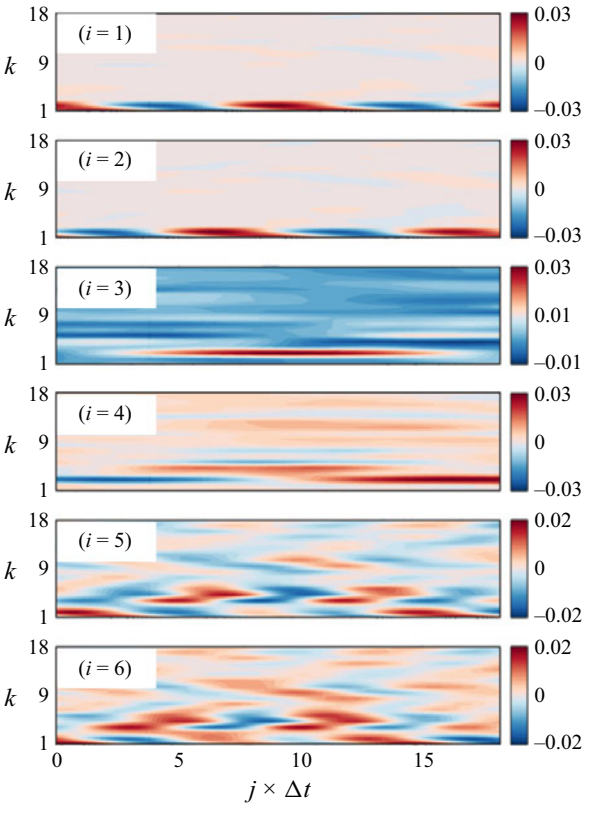


Figure 17. Contours of the left singular vectors $\boldsymbol{U}_{H,i}^{(u,v,w)}$ in the time-delay/mode-order plane, for $m_u = 18$, $q = 956$, $q \times \Delta t = 18.16$.

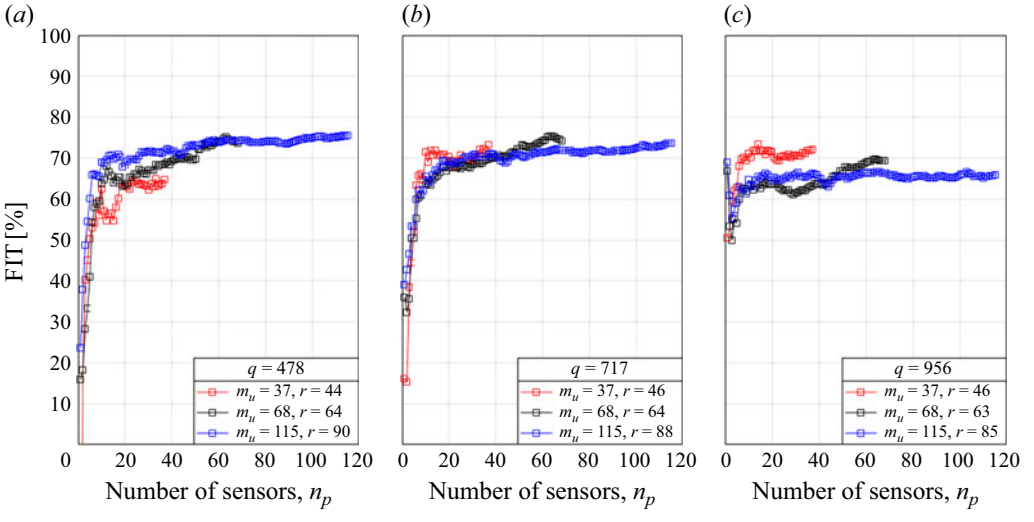


Figure 18. The FIT[%] metric against the number of velocity sensors, n_p: (*a*) $q \times \Delta t = 9.08$; (*b*) $q \times \Delta t = 13.62$; (*c*) $q \times \Delta t = 18.16$.

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

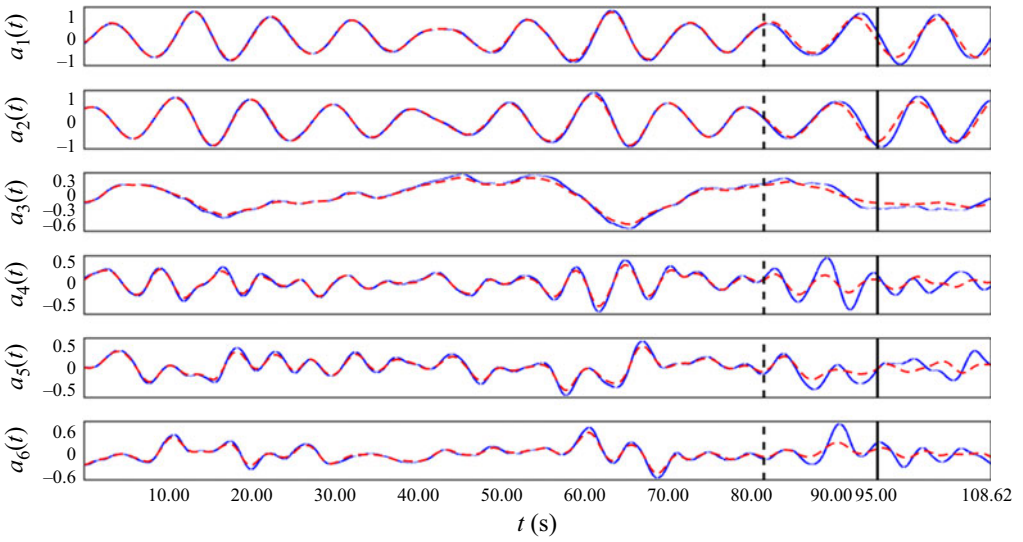


Figure 19. Forecasting of the future evolution of the POD coefficients using velocity measurements for $q \times \Delta t = 13.62$ with $m_u = 68$, $n_p = 63$. The solid vertical line indicates the starting point of the forecasting dataset. The time window between the dashed and solid lines marks t_p to t_{Ktrain}, which is the time extent of the last column of the Hankel matrix $\boldsymbol{H}$. Blue lines indicate DNS and red lines reconstruction/forecasting.

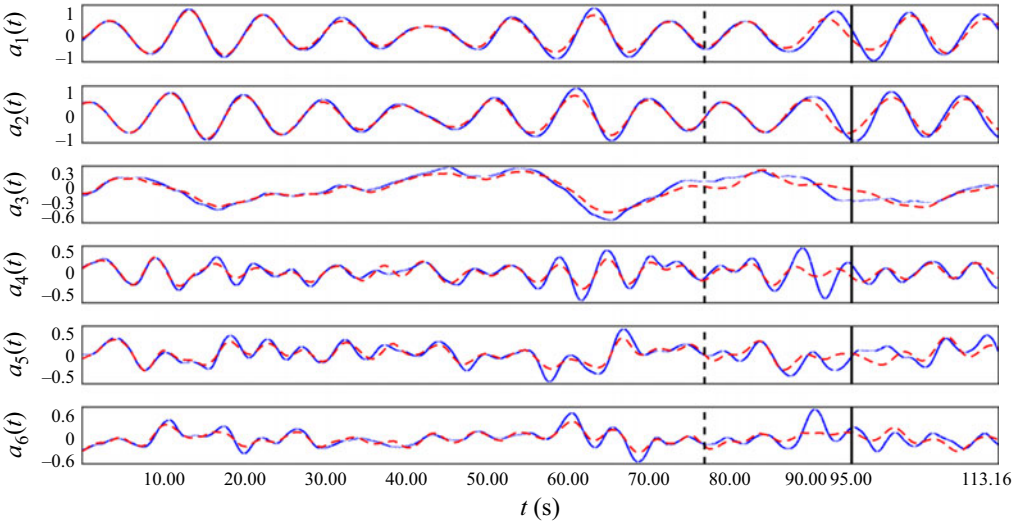


Figure 20. Forecasting of the future evolution of the POD coefficients using velocity measurements for $q \times \Delta t = 18.16$ with $m_u = 37$, $n_p = 14$. Blue lines indicate DNS and red lines reconstruction/forecasting. For the meaning of the vertical dashed and solid lines, refer to the caption of figure 19.

Figures 19 and 20 depict the reconstruction of time coefficients of the six dominant POD modes in the training dataset and the future evolution in the forecasting dataset for the two largest time widows $q \times \Delta t = 13.62$ and 18.16, respectively. The process to obtain these figures is as follows. Velocity measurements $\boldsymbol{s}[k]$ at n_p points are employed to estimate the time coefficients $\boldsymbol{a}[k]$ from $k = 0 \ldots K_{train}$. At $k = K_{train}$, the estimation is extended from $k_{train+1} \ldots K_{train} + q$. Given the three-dimensional turbulent nature of the flow, the

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

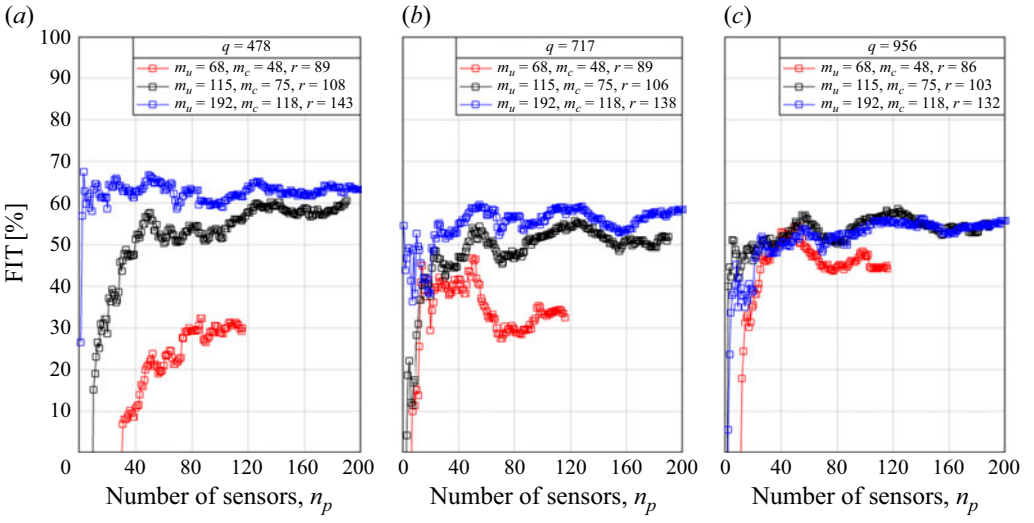


Figure 21. The FIT[%] metric against the number of scalar sensors, n_p: (*a*) $q \times \Delta t = 9.08$; (*b*) $q \times \Delta t = 13.62$; (*c*) $q \times \Delta t = 18.16$.

forecasting quality of the first two dominant modes is impressive. Both the amplitude and phase are well predicted even for the largest time delay (which is more than two orders of magnitude larger than the Lyapunov time scale). The slowly evolving third mode is also well forecast. Small discrepancies are detected in the higher modes, but again the phase and amplitude are satisfactorily reproduced especially for the largest time window. Recall that modes 4−6 have very little energy, in the range (1−2) % as shown in figure 9, so these deviations are not surprising. The estimated $\boldsymbol{a}[k]$ between $k = K_{train} - q \ldots K_{train}$ (region between dashed and solid lines) slightly deviates from the ground truth, probably because this is the last column in the Hankel matrix $\boldsymbol{H}$. These small deviations are more pronounced for modes 4−6 and also propagate inside the forecasting window affecting the accuracy. Note that we obtained the results of figures 19 and 20 using the correct initial value of $\hat{\boldsymbol{v}}_H[0]$ (for this reason the blue and red lines match at $t = 0$). However, this is not a restriction. If the value of $\hat{\boldsymbol{v}}_H[0]$ is not known, it can be set as 0. In this case, $\hat{\boldsymbol{v}}_H[k]$ will be updated as streaming data $\boldsymbol{s}[k]$ are coming in, and the estimated $\hat{a}_i[k]$ will take some time to catch up with the true solution $a_i[k]$.

4.5. *Flow-field reconstruction and forecasting from scalar measurements*

We now turn our attention to the forecasting of the velocity field from scalar measurements. Recall that we place the scalar sensors at the peaks of the POD velocity modes, but they record only one piece of information, the concentration.

In figure 21, we explore the effect of the number of velocity and scalar POD modes m_u and m_c, respectively, and the number of measurements n_p on the FIT[%] metric for the same values of time-delayed snapshots q as in figure 18. It can be seen that now larger values of m_u and m_c are required for the results to converge. As in the case of velocity measurements, FIT[%] is reduced, but only slightly, as q increases. Again this is a positive result that demonstrates that one can robustly forecast the velocity field from scalar measurements which are more cost effective to obtain. The FIT[%] values are slightly smaller compared with the results of the previous section. Most likely this is because the sensors record only the scalar concentration, i.e. they provide less information

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

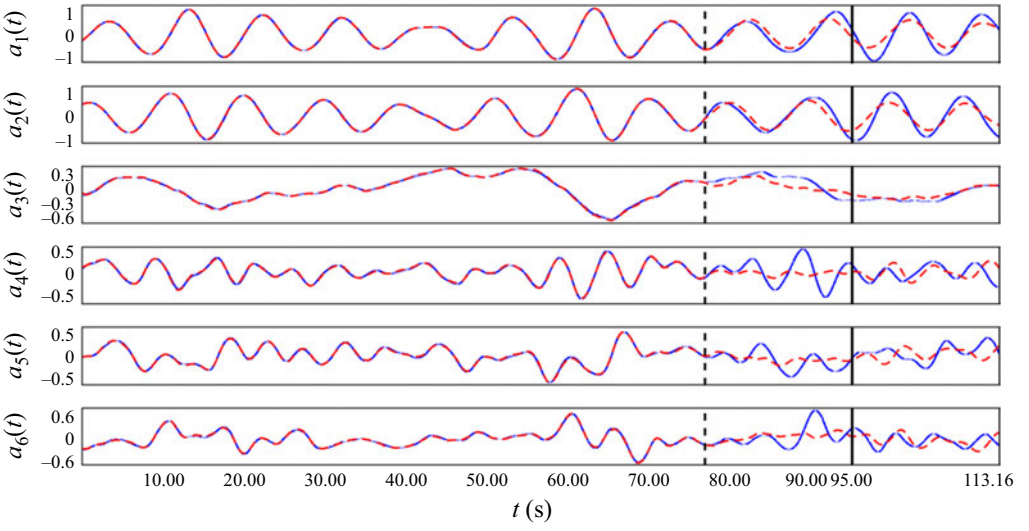


Figure 22. Forecasting of the future evolution of the dominant POD coefficients using scalar measurements for $q \times \Delta t = 18.16$ with $m_u = 115$, $m_c = 75$ and $n_p = 123$.

to the estimator. The performance of reconstruction from a small number of sensors again improves as the time delay $q \times \Delta t$ grows, but the improvement is not as clear as when velocity sensors are used.

In figure 22 we plot the reconstructed and forecast time series of the first six POD modes from scalar measurements for the largest $q \times \Delta t = 18.16$. Again, the time signals closely follow the true ones in the training dataset. In the forecasting dataset, the model predicts the first three time coefficients very well. The quality is comparable to that of figure 20 where velocity sensors are used. These plots demonstrate that it is possible to get accurate results from scalar measurements.

5. Conclusions

A data-driven estimator was synthesised that can forecast the future evolution of a three-dimensional turbulent flow field from current sparse velocity and/or scalar measurements. This is made possible by combining time-delayed embedding, Koopman theory and optimal estimation theory. The key idea is the construction of a linear dynamical system that governs the future POD coefficients and closure of the system using sensor measurements.

The estimator was applied to the three-dimensional turbulent recirculating flow around a surface-mounted cube. Velocity (and scalar, if required) sensors were placed at the peaks of the velocity POD modes. Forecasting the future evolution of the dominant POD modes was performed over time windows one to two orders of magnitude larger than the Lyapunov time scale. Accurate forecasting was obtained for the first three velocity POD modes. Results with velocity sensors were slightly more accurate compared with scalar sensors. The forecasting accuracy only slightly decreased as the window size increased. This was true for both velocity and scalar sensors. The results demonstrate that long forecasts can be made for the most dominant structures of the flow.

The method can be extended to include other dimensionality reduction techniques, for example convolutional autoencoders. This will allow more compact representation of

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

the dynamics, which is especially useful for high-Reynolds-number flows. The resulting nonlinearity of the mapping between the latent space variables and the velocity field can be dealt with using other estimators, such as the ensemble Kalman filter. Application of the method to other flow settings, such as wall-bounded flows using pressure and/or shear stress measurements, would be particularly interesting. The framework can also easily incorporate data from moving sensors, such as drones or wearable sensors. Another extension is to train at different operating conditions (e.g. different Reynolds members) and forecast the flow at an unseen condition using only streaming data from the new condition.

Acknowledgements. The authors would like to thank EPSRC for the computational time made available on the UK supercomputing facility ARCHER/ARCHER2 via the UK Turbulence Consortium (EP/X035484/1).

Funding. G.P. is supported by EPSRC grants EP/X017273/1 and EP/W001748/1.

Declaration of interests. The authors report no conflict of interest.

Author contributions. Both authors contributed equally to analysing data and reaching conclusions, and in writing the paper.

References

Allen, A., Markou, S. & Tebbutt, W. 2025 End-to-end data-driven weather prediction. *Nature* **641**, 1172–1179.

Arbabi, H. & Mezić, I. 2017 Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the koopman operator. *SIAM J. Appl. Dyn. Syst.* **16** (4).

Bourgeois, J.A., Sattari, P. & Martinuzzi, R.J. 2011 Alternating half-loop shedding in the turbulent wake of a finite surface-mounted square cylinder with a thin boundary layer. *Phys. Fluid* **23**, 095101.

Box, G.E.P., Jenkins, G.M., Reinsel, G.C. & Ljung, G.M. 2015 *Time Series Analysis: Forecasting and Control*, 5th edn. Wiley.

Brunton, S.L. & Noack, B.R. 2015 Closed-loop turbulence control: progress and challenges. *Appl. Mech. Rev.* **67** (5), 050801.

Brunton, S.L., Brunton, B.W., Proctor, J.L., Kaiser, E. & Kutz, J.N. 2017 Chaos as an intermittently forced linear system. *Nat. Commun.* **8** (19).

Brunton, S.L., Budišić, M., Kaiser, E. & Kutz, J.N. 2022 Modern koopman theory for dynamical systems. *SIAM Rev.* **64** (2), 229–340.

Callaham, J.L., Maeda, K. & Brunton, S.L. 2019 Robust flow reconstruction from limited measurements via sparse representation. *Phys. Rev. Fluids* **4**, 103907.

Castro, I.P. & Robins, A.G. 1977 The flow around a surface-mounted cube in uniform and turbulent streams. *J. Fluid Mech.* **79**, 307–335.

Chu, T. & Schmidt, O.T. 2025 Stochastic reduced-order koopman model for turbulent flows. *Proc. R. Soc. A: Math. Phys. Engng Sci.* **481**(2323).

Constante-Amores, C.R., Linot, A.J. & Graham, M.D. 2025 Dynamics of a data-driven low-dimensional model of turbulent minimal pipe flow. *J. Fluid Mech.* **1020**, A58.

Crisanti, A., Jensen, M.H., Vulpiani, A. & Paladin, G. 1993 Intermittency and predictability in turbulence. *Phys. Rev. Lett.* **70**, 166–169.

Das, S. & Giannakis, D. 2019 Delay-coordinate maps and the spectra of koopman operators. *J. Stat. Phys.* **175**, 1107–1145.

Dubois, P., Gomez, T., Planckaert, L. & Perret, L. 2020 Data-driven predictions of the Lorenz system. *Physica D* **408**, 132495.

Dylewsky, D., Kaiser, E., Brunton, S.L. & Kutz, J.N. 2022 Principal component trajectories for modeling spectrally continuous dynamics as forced linear systems. *Phys. Rev. E* **105**, 015312.

Eivazi, H., Guastoni, L., Schlatter, P., Azizpour, H. & Vinuesa, R. 2021 Recurrent neural networks and Koopman-based frameworks for temporal predictions in a low-order model of turbulence. *Intl J. Heat Fluid Flow* **90**, 108816.

Evensen, G. 2003 The ensemble kalman filter: theoretical formulation and practical implementation. *Ocean Dyn.* **53**, 343–367.

Frame, P. & Towne, A. 2023 Space-time POD and the Hankel matrix. *PLOS One* **18** (8), 1–31.

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

GE, J., ROLLAND, J. & VASSILICOS, J.C. 2023 The production of uncertainty in three-dimensional Navier–Stokes turbulence. *J. Fluid Mech.* **977**, A17.

GIANNAKIS, D. 2019 Data-driven spectral decomposition and forecasting of ergodic dynamical systems. *Appl. Comput. Harmon. Anal.* **47** (2), 338–396.

HASSANALY, M. & RAMAN, V. 2019 Lyapunov spectrum of forced homogeneous isotropic turbulent flows. *Phys. Rev. Fluids* **4** (11), 114608.

KAILATH, T., HASSIBI, B. & SAYED, A.H. 2000 *Linear Estimation*. Prentice-Hall International.

KAMB, M., KAISER, E., BRUNTON, S.L. & KUTZ, J.N. 2020 Time-delay observables for Koopman: theory and applications. *SIAM J. Appl. Dyn. Syst.* **19** (2), 886–917.

KARNIADAKIS, G.E., KEVREKIDIS, I.G., LU, L., PERDIKARIS, P., WANG, S. & YANG, L. 2021 Physics-informed machine learning. *Natl Rev. Phys.* **3**, 422–440.

KHODKAR, M.A. & HASSANZADEH, P. 2021 A data-driven, physics-informed framework for forecasting the spatiotemporal evolution of chaotic dynamics with nonlinearities modeled as exogenous forcings. *J. Comput. Phys.* **440**, 110412.

KNIGHT, B. & SIROVICH, L. 1990 Kolmogorov inertial range for inhomogeneous turbulent flows. *Phys. Rev. Lett.* **65** (11), 1356.

KRAJNOVIC, S. & DAVIDSON, L. 2002 Large-eddy simulation of the flow around a bluff body. *AIAA J.* **40**, 927–936.

LE CLAINCHE, S. & VEGA, J.M. 2017 Higher order dynamic mode decomposition. *SIAM J. Appl. Dyn. Syst.* **16** (2), 882–925.

LI, X., HULSHOFF, S. & HICKEL, S. 2021 Towards adjoint-based mesh refinement for large eddy simulation using reduced-order primal solutions: preliminary 1d burgers study. *Comput. Meth. Appl. Mech. Engng* **379**, 113733.

LORENZ, E.N. 1963 Deterministic non-periodic flows. *J. Atmos. Sci.* **20**, 130–141.

MEZIĆ, I. 2013 Analysis of fluid flows via spectral properties of the koopman operator. *Annu. Rev. Fluid Mech.* **45**, 357–378.

MOEHLIS, J., FAISST, H. & ECKHARDT, B. 2004 A low-dimensional model for turbulent shear flows. *New J. Phys.* **6** (1), 56.

MOHAN, P., FITZSIMMONS, N. & MOSER, R.D. 2017 Scaling of Lyapunov exponents in homogeneous isotropic turbulence. *Phys. Rev. Fluids* **2**, 114606.

OTTO, S.E. & ROWLEY, C.W. 2021 Koopman operators for estimation and control of dynamical systems. *Ann. Rev. Control Robotics Auton. Syst.* **4** (2021), 59–87.

PAN, S. & DURAISAMY, K. 2019 On the structure of time-delay embedding in linear models of non-linear dynamical systems. *Chaos* **30** (7), 073135.

PAN, S. & DURAISAMY, K. 2020 Physics-informed probabilistic learning of linear embeddings of nonlinear dynamics with guaranteed stability. *SIAM J. Appl. Dyn. Syst.* **19** (1), 480–509.

PATHAK, J., LU, Z., HUNT, B.R., GIRVAN, M. & OTT, E. 2017 Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data. *Chaos: Interdiscip. J. Nonlinear Sci.* **27** (12), 121102.

POPE, S.B. 2000 *Turbulent Flows*. Cambridge University Press.

ROSSI, R., PHILIPS, D.A. & IACCARINO, G. 2010 A numerical study of scalar dispersion downstream of a wall-mounted cube using direct simulations and algebraic flux models. *Intl J. Heat Fluid Flow* **31**, 805–819.

RUELLE, D. 1979 Microscopic fluctuations and turbulence. *Phys. Lett. A* **72** (2), 81–82.

SCHLANDER, R.K., RIGOPOULOS, S. & PAPADAKIS, G. 2024 Resolvent analysis of turbulent flow laden with low-inertia particles. *J. Fluid Mech.* **985**, A27.

SCHMID, P.J. 2010 Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* **656**, 5–28.

SCHMIDT, O.T. 2026 Data-driven forecasting of high-dimensional transient and stationary processes via space–time projection. *Proc. R. Soc. A: Math. Phys. Engng Sci.* **482** (2329), 20250454.

SIPP, D. & SCHMID, P.J. 2016 Linear closed-loop control of fluid instabilities and noise-induced perturbations: a review of approaches and tools. *Appl. Mech. Rev.* **68** (2), 020801.

SIROVICH, L. 1987 Turbulence and the dynamics of coherent structures. *Q. Appl. Maths* **45**, 561–571.

TAKENS, F. 1981 Detecting Strange Attractors in Turbulence. In *Dynamical Systems and Turbulence, Warwick 1980* (ed. R. David & L.-S. Young), pp. 366–381. Springer.

THOMAREIS, N. & PAPADAKIS, G. 2017 Effect of trailing edge shape on the separated flow characteristics around an airfoil at low Reynolds number: a numerical study. *Phys. Fluids* **29** (1), 014101.

THOMAREIS, N. & PAPADAKIS, G. 2018 Resolvent analysis of separated and attached flows around an airfoil at transitional Reynolds number. *Phys. Rev. Fluids* **3**, 073901.

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press

VLACHAS, P.R., PATHAK, J., HUNT, B.R., SAPSIS, T.P., GIRVAN, M., OTT, E. & KOUMOUTSAKOS, P. 2020 Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks* **126**, 191–217.

YAO, H. & PAPADAKIS, G. 2023 On the role of the laminar/turbulent interface in energy transfer between scales in bypass transition. *J. Fluid Mech.* **960**, A24.

ZAKI, T.A. 2025 Turbulence from an observer perspective. *Annu. Rev. Fluid Mech.* **57**, 311–334.

ZHANG, F., SUN, Y.Q., MAGNUSSON, L., BUIZZA, R., LIN, S.-J., CHEN, J.-H. & EMANUEL, K. 2019 What is the predictability limit of midlatitude weather? *J. Atmos. Sci.* **76** (4), 1077–1091.

https://doi.org/10.1017/jfm.2026.11251 Published online by Cambridge University Press