


Data-assisted, physics-informed propagators for recurrent flows

T. Lichtenegger ^{*}

*Department of Particulate Flow Modelling, Johannes Kepler University Linz,
Altenberger Strasse 69, 4040 Linz, Austria*



(Received 18 July 2023; accepted 29 January 2024; published 20 February 2024)

An alternative approach to simulate dynamic, recurrent flows with very large time steps is presented. Data-driven forecasts based on the method of analogs are made employing a set of precomputed time series segments of flow fields. These predictions are then corrected with the evolved deviation between initial flow state and its nearest neighbor in the database. The evolution of these deviations is obtained from the convolution with the propagator of the linearized Navier-Stokes equations, where linearization takes place around the most similar state in the database. The error E_{lin} due to this approximation implies the appropriate distance function to determine nearest-neighbor fields. We tested the method on the simple case of 2D vortex shedding behind a cylinder at Reynolds numbers in the range of 200 with step sizes of up to 200 times larger than those used to generate the underlying data with the PISO algorithm. Short-term forecasts showed a high degree of accuracy with errors correlating well with E_{lin} for a number of case variations. But also after many steps, the predicted flow fields remained in close agreement with high-fidelity CFD calculations with relative errors on the order of 1% for the temporal averages. Furthermore, excellent conservation behavior of global properties like total momentum and kinetic energy was observed.

DOI: [10.1103/PhysRevFluids.9.024401](https://doi.org/10.1103/PhysRevFluids.9.024401)

I. INTRODUCTION

Transient fluid mechanical problems and their numerical treatment are connected to small time steps with expensive calculations in each of them. Even though hardware resources have increased significantly to allow for massively parallel computations, it is still very demanding to simulate highly dynamic flows over long durations. For this reason, temporal multiscale problems involving very fast and slow processes have remained a serious challenge.

Over the past few years, data-driven numerical techniques involving elements of machine learning (ML) have started to enter fluid mechanics. An overview of the vast number of methods and their applications can be found in the review of Brunton *et al.* [1]. Various strategies, many of them built upon deep neural networks (NNs), to learn certain aspects of flow dynamics from data were devised. Generative adversarial networks (GANs) were trained to make increasingly realistic predictions about the evolution [2–4] and about subresolution structures [5–7] of flows. Others extracted characteristic patterns and reduced the dimensionality of the problem to derive reduced-order models (ROMs). Besides the classical combination of proper orthogonal decomposition (POD) with a Galerkin approach [8–11], more recent works employed autoencoders [12–17] to allow for a nonlinear decomposition. Equations of motion (EOMs) acting in these reduced coordinates [18–21] come with significantly lower computational costs than those of the original problem.

^{*}thomas.lichtenegger@jku.at

The impact of combining data with the underlying laws of physics was demonstrated by Raissi *et al.* [22–24] in terms of physics-informed neural networks (PINNs), which have proven to be extremely powerful tools for ill-posed and inverse problems and have been applied to a wide range of topics [25–29]. Including the governing EOMs into the loss function of a NN allowed to make accurate predictions with comparatively little data as opposed to techniques which are not aware of any physical constraints. An overview of the rapidly growing field of physics-informed ML is provided by the review of Karniadakis *et al.* [30].

In stark contrast to such elaborate techniques, we developed the deliberately simple yet very fast method recurrence CFD (rCFD) specifically targeted at temporal multiscale problems in recurrent flows like fluidized beds [31] or turbulent jets [32]. Using a database of prerecorded field time series, rCFD performs large-step predictions by iterating the method of analogs [33]. This way, flow sequences of almost arbitrary length can be easily generated, which can serve as a dynamic background for slow long-term processes to take place. Especially for passive transport, rCFD enabled extremely fast simulations several orders of magnitude [34] faster than the underlying CFD calculations after the initial construction of the database with said methods. Of course, rCFD does not reach the same level of accuracy as classic CFD methods but can provide results in semi-quantitative agreement with them. Conversely, standard CFD algorithms can be accelerated to some degree by using larger steps and decreasing the required accuracy, but could hardly achieve similar CPU times because they would still have to solve the coupled velocity and pressure equations. However, it cannot be denied that open, challenging questions for rCFD persist such as the handling of high Reynolds number flows, where the velocity field varies over timescales beyond the scope of a single database [35].

For the case of long-term simulations, it is particularly important but also challenging to ensure that the governing laws of physics are still satisfied after long durations. While physics-informed ML models include the EOMs directly in their training, rCFD uses only flow fields from its physically valid database. Therefore, rCFD is more or less bound to the conditions at which the database was created, but at the same time does not run into any danger of leaving the underlying attractor even after a huge number of steps. Physics-informed ML techniques, however, can offer more flexibility but might struggle with multiscale problems [30,36].

In this work, we propose an alternative approach that combines the benefits of flexibility and stability in a clearly interpretable fashion. Large-step predictions are made based on the method of analogs. If the exactly same state as that for which the forecast is to be computed can be found in the database, then one could infer the evolution without any uncertainty. However, in general there will be deviations between the state under consideration and its nearest neighbor from the database, e.g., due to different boundary conditions or changes of internal parameters. Clearly, these differences need to be taken into account to correct the predictions. To this end, we derive equations for the propagation of the deviations by linearizing the full, nonlinear Navier-Stokes equations around the nearest neighbor state. In contrast to (partly) implicit techniques like backward differentiation formulas (BDF) or explicit singly diagonal implicit Runge-Kutta (ESDIRK) methods [37–39], which can tolerate large time steps, too, the present algorithm is formulated in an almost completely explicit fashion (cf. Sec. IV D), which gives it the potential for much faster simulations.

Related ideas were put forward by Cacuci *et al.* [40,41], who developed a Green’s function formalism for nonlinear problems. However, they obtained self-consistent integral equations whereas we restrict ourselves to a first-order expansion that leads to a simple, explicit scheme in terms of a deviation propagator. Gin *et al.* [42] used autoencoders to transform boundary value problems into linear coordinates where a Green’s function can be found, but a physics-informed extension to time-dependent settings has not been attempted yet. Boullé *et al.* [43] used NNs to learn the response of a system excited with a Gaussian process. Their approach may also be applied to nonlinear EOMs to obtain the Green’s function for an approximate linear model. It is an open question to which extent this simplification can account for a nonlinearity in transient problems over long durations. Similarly, physics-informed deep operator networks [44–47] advance a flow state in time by learning the underlying, nonlinear EOMs, which becomes increasingly challenging

for large, multiscale systems. In contrast, our method allows for a clean separation into a nonlinear subproblem and a linear one for which a propagator can be computed, but relies on a database of similar flow fields. Therefore, it is applicable when similar states are repeatedly encountered. This comprises transio-recurrent dynamics and cases where the effects of parameter variation, e.g., for the purpose of optimization, are of interest. We remark that the combination of nearest-neighbor prediction with deviation propagation may also be applied to other nonlinear problems than the Navier-Stokes equations.

The paper is organized as follows: The theoretical basis of the deviation-propagation formalism is explained in Sec. II. In Sec. III, we describe the test case that serves to illustrate our method with results presented in Sec. IV. Finally, we draw conclusions and discuss limitations in Sec. V.

II. THEORETICAL BACKGROUND

A. Linearized propagator ansatz

We investigate the dynamics of an incompressible fluid of viscosity ν with velocity $\mathbf{u}(\mathbf{r}, t)$ and pressure $p(\mathbf{r}, t)$ in a domain Ω governed by the Navier-Stokes equations

$$\nabla \cdot \mathbf{u} = 0, \quad (1)$$

$$\frac{\partial}{\partial t} \mathbf{u} + \nabla \cdot \mathbf{u} \mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u}, \quad (2)$$

here written in conservative form. Given two sufficiently similar flow fields with sufficiently similar values on the boundaries, which are subject to the same EOMs, it is reasonable to assume that they will evolve in a similar fashion for some (short) time period. More specifically, assuming that the evolution of a reference state $\mathbf{u}_{\text{ref}}(\mathbf{r}, 0 < t' \leq t)$, $p_{\text{ref}}(\mathbf{r}, 0 < t' \leq t)$ is known up to time t and available in a database of previously calculated time series segments, we want to model the dynamics observed for similar initial conditions

$$\mathbf{u}(\mathbf{r}, 0) = \mathbf{u}_{\text{ref}}(\mathbf{r}, 0) + \delta \mathbf{u}(\mathbf{r}, 0), \quad (3)$$

$$p(\mathbf{r}, 0) = p_{\text{ref}}(\mathbf{r}, 0) + \delta p(\mathbf{r}, 0), \quad (4)$$

with similar boundary values. For the sake of simplicity, we assume that the boundary of Ω , $\partial\Omega = \partial\Omega_u \cup \partial\Omega_p$, consists of patches $\partial\Omega_u$ with fixed value of velocity and gradient of pressure and of patches $\partial\Omega_p$ with fixed value of pressure and gradient of velocity so that

$$\left. \begin{aligned} \mathbf{u}(\mathbf{r}_b, t) &= \mathbf{u}_{\text{ref}}(\mathbf{r}_b, t) + \delta \mathbf{u}(\mathbf{r}_b, t) \\ \mathbf{n} \cdot \nabla p(\mathbf{r}_b, t) &= \mathbf{n} \cdot \nabla p_{\text{ref}}(\mathbf{r}_b, t) + \delta p'(\mathbf{r}_b, t) \end{aligned} \right\} \text{ for } \mathbf{r}_b \in \partial\Omega_u, \quad (5)$$

and

$$\left. \begin{aligned} \mathbf{n} \cdot \nabla \mathbf{u}(\mathbf{r}_b, t) &= \mathbf{n} \cdot \nabla \mathbf{u}_{\text{ref}}(\mathbf{r}_b, t) + \delta \mathbf{u}'(\mathbf{r}_b, t) \\ p(\mathbf{r}_b, t) &= p_{\text{ref}}(\mathbf{r}_b, t) + \delta p(\mathbf{r}_b, t) \end{aligned} \right\} \text{ for } \mathbf{r}_b \in \partial\Omega_p. \quad (6)$$

As long as all deviations indicated by a leading δ are small, we can make a linear ansatz

$$\mathbf{u}(\mathbf{r}, t) = \mathbf{u}_{\text{ref}}(\mathbf{r}, t) + \int_{\Omega} d^3 r' \mathbf{K}_{uu}(\mathbf{r}, \mathbf{r}', t) \cdot \delta \mathbf{u}(\mathbf{r}', 0) + \int_{\Omega} d^3 r' \mathbf{K}_{up}(\mathbf{r}, \mathbf{r}', t) \delta p(\mathbf{r}', 0), \quad (7)$$

$$p(\mathbf{r}, t) = p_{\text{ref}}(\mathbf{r}, t) + \int_{\Omega} d^3 r' \mathbf{K}_{pu}(\mathbf{r}, \mathbf{r}', t) \cdot \delta \mathbf{u}(\mathbf{r}', 0) + \int_{\Omega} d^3 r' \mathbf{K}_{pp}(\mathbf{r}, \mathbf{r}', t) \delta p(\mathbf{r}', 0), \quad (8)$$

where integration is carried out over the whole domain Ω including the boundaries. The latter part is essential because of delta functions located on them (cf. the following sections).

While the first term on the right-hand sides of Eqs. (7) and (8) represents the evolution of the reference state, the others account for the effect of the deviations of velocity and pressure. The kernels \mathbf{K}_{uu} , \mathbf{K}_{up} , \mathbf{K}_{pu} , and \mathbf{K}_{pp} can be regarded as yet to be determined components of the

propagator that evolves a small deviation $\delta \mathbf{u}(0)$, $\delta p(0)$ from a reference state over a duration t into $\delta \mathbf{u}(t)$, $\delta p(t)$. They are closely related to the Green's function of the system (cf. Sec. II C).

As long as the database is sufficiently extensive, a similar reference state for $\mathbf{u}(t)$, $p(t)$ computed by Eqs. (7) and (8) can be found and the procedure repeated to obtain $\mathbf{u}(2t)$, $p(2t)$ and so on.

B. Initial and boundary conditions of the propagators

The initial conditions

$$\mathbf{K}_{uu}(\mathbf{r}, \mathbf{r}', 0) = \mathbf{1}\delta(\mathbf{r} - \mathbf{r}'), \quad (9)$$

$$\mathbf{K}_{up}(\mathbf{r}, \mathbf{r}', 0) = \mathbf{0} \quad (10)$$

follow from the requirement that $\mathbf{u}(\mathbf{r}, t \rightarrow 0) = \mathbf{u}_{\text{ref}}(\mathbf{r}, t \rightarrow 0) + \delta \mathbf{u}(\mathbf{r}, 0)$ in Eq. (7). No initial conditions are needed for \mathbf{K}_{pu} and \mathbf{K}_{pp} because no time derivatives of pressure are computed in incompressible flow.

In addition to the initial conditions, one has to define appropriate boundary conditions. They need to be specified only for the first coordinate of the propagators because no spatial derivatives with respect to the second coordinates are calculated (cf. Sec. II C). If \mathbf{r} approaches a point \mathbf{r}_b on the boundary, then one finds that

$$\left. \begin{aligned} \mathbf{K}_{uu}(\mathbf{r} \rightarrow \mathbf{r}_b, \mathbf{r}') &= \mathbf{1}\delta(\mathbf{r}_b - \mathbf{r}') \\ \mathbf{K}_{up}(\mathbf{r} \rightarrow \mathbf{r}_b, \mathbf{r}') &= \mathbf{0}, \end{aligned} \right\} \text{ for } \mathbf{r}_b \in \partial\Omega_u \quad (11)$$

and

$$\left. \begin{aligned} \mathbf{n} \cdot \nabla \mathbf{K}_{uu}(\mathbf{r} \rightarrow \mathbf{r}_b, \mathbf{r}') &= \mathbf{n} \cdot \nabla \mathbf{1}\delta(\mathbf{r}_b - \mathbf{r}') \\ \mathbf{n} \cdot \nabla \mathbf{K}_{up}(\mathbf{r} \rightarrow \mathbf{r}_b, \mathbf{r}') &= \mathbf{0} \end{aligned} \right\} \text{ for } \mathbf{r}_b \in \partial\Omega_p. \quad (12)$$

have to hold such that the velocity field according to ansatz Eq. (7) satisfies the boundary conditions Eqs. (5) and (6). Similarly, if the pressure components of the propagator obey

$$\left. \begin{aligned} \mathbf{n} \cdot \nabla \mathbf{K}_{pu}(\mathbf{r} \rightarrow \mathbf{r}_b, \mathbf{r}') &= \mathbf{0} \\ \mathbf{n} \cdot \nabla \mathbf{K}_{pp}(\mathbf{r} \rightarrow \mathbf{r}_b, \mathbf{r}') &= \mathbf{n} \cdot \nabla \delta(\mathbf{r}_b - \mathbf{r}') \end{aligned} \right\} \text{ for } \mathbf{r}_b \in \partial\Omega_u \quad (13)$$

and

$$\left. \begin{aligned} \mathbf{K}_{pu}(\mathbf{r} \rightarrow \mathbf{r}_b, \mathbf{r}') &= \mathbf{0} \\ \mathbf{K}_{pp}(\mathbf{r} \rightarrow \mathbf{r}_b, \mathbf{r}') &= \delta(\mathbf{r}_b - \mathbf{r}') \end{aligned} \right\} \text{ for } \mathbf{r}_b \in \partial\Omega_p, \quad (14)$$

then the predicted pressure field agrees with the prescribed boundary conditions.

C. Propagator EOMs

For consistency reasons, the linear ansatz Eqs. (7) and (8) implies that the propagators are determined from the linearized Navier-Stokes equations, where linearization of Eqs. (1) and (2) takes place around $\mathbf{u}_{\text{ref}}(\mathbf{r}, t)$, $p_{\text{ref}}(\mathbf{r}, t)$,

$$\nabla \cdot \delta \mathbf{u} = 0, \quad (15)$$

$$\frac{\partial}{\partial t} \delta \mathbf{u} + \nabla \cdot \delta \mathbf{u} \mathbf{u}_{\text{ref}} + \nabla \cdot \mathbf{u}_{\text{ref}} \delta \mathbf{u} = -\nabla \delta p + \nu \nabla^2 \delta \mathbf{u}. \quad (16)$$

This way, we have moved all the complexity due to the nonlinearity of the full Navier-Stokes equations into the creation of a representative database of flow field time series segments. Any deviation from these prerecorded states is dealt with by the propagators in a linearized fashion.

In the above approximation, one neglects the second-order term $\nabla \cdot \delta \mathbf{u} \delta \mathbf{u}$. This is valid as long as its magnitude satisfies

$$|\nabla \cdot \delta \mathbf{u} \delta \mathbf{u}| \ll |\nabla \cdot \delta \mathbf{u} \mathbf{u}_{\text{ref}} + \nabla \cdot \mathbf{u}_{\text{ref}} \delta \mathbf{u}|, \quad (17)$$

which makes it a reasonable indicator to identify a sufficiently similar reference state in terms of

$$E_{\text{lin}} \equiv \int_{\Omega} d^3r |\nabla \cdot \delta \mathbf{u} \delta \mathbf{u}|. \quad (18)$$

Furthermore, Eq. (17) allows to estimate an upper bound for the spacing of reference states. If $\mathbf{u}(\mathbf{r}, t_n)$ is the n th reference velocity field, then the next one needs to be added to the database before $\delta \mathbf{u} = \mathbf{u}(\mathbf{r}, t_{n+1}) - \mathbf{u}(\mathbf{r}, t_n)$ and $\mathbf{u}_{\text{ref}} = \mathbf{u}(\mathbf{r}, t_n)$ violate Eq. (17). Of course, this criterion is only sufficient if the large-step simulations are carried out at the conditions of the database. For different boundary conditions or internal parameter values, $\delta \mathbf{u}$ cannot be made arbitrarily small by refining the distance between reference states. Instead, its magnitude is limited from below by the distance between the current conditions and those of the database. For such cases, Eq. (17) may be repeatedly checked during large-step simulations to assess the validity of the propagator approach.

If we now insert the ansatz functions and keep in mind that $\delta \mathbf{u}(\mathbf{r}, 0)$ and $\delta p(\mathbf{r}, 0)$ are independent from each other, then we obtain separate equations for the propagation of velocity variations \mathbf{K}_{uu} , \mathbf{K}_{pu} and for the propagation of pressure variations \mathbf{K}_{up} , \mathbf{K}_{pp} .

First, we deal with the former case. For Eq. (7) to satisfy the continuity equation, one needs

$$\int_{\Omega} d^3r' \nabla \cdot \mathbf{K}_{uu}(\mathbf{r}, \mathbf{r}', t) \cdot \delta \mathbf{u}(\mathbf{r}', 0) = 0. \quad (19)$$

Although it would be tempting to assume $\nabla \cdot \mathbf{K}_{uu}(\mathbf{r}, \mathbf{r}', t) = 0$, this equation is clearly invalid as it conflicts with the initial condition Eq. (9). The reason for this seeming contradiction lies in the fact that $\mathbf{K}_{uu}(\mathbf{r}, \mathbf{r}', t)$ is defined to only operate on physically valid, solenoidal states $\delta \mathbf{u}(\mathbf{r}, 0)$. Hence, one can conclude from Eq. (19) only that $\nabla \cdot \mathbf{K}_{uu}(\mathbf{r}, \mathbf{r}', t) = \mathcal{N}(\mathbf{r}, \mathbf{r}', t)$ with some operator \mathcal{N} the evaluates to 0 if applied to a divergence-free flow field.

In this context, it might be advantageous to expand $\mathbf{K}_{uu}(\mathbf{r}, \mathbf{r}', t)$ in a basis of divergence-free finite elements [48–51] with respect to \mathbf{r}' where the \mathbf{r} -dependent coefficients would be indeed solenoidal. Upon representing the velocity field in the same basis, large-step predictions would be divergence-free per construction. However, in the present investigation we restrict ourselves to the simpler finite-volume discretization, for which we first derive the pressure equation from Eqs. (15) and (16) and then eliminate any terms of the form $\nabla \cdot \delta \mathbf{u}(\mathbf{r}, t)$ (which are identical 0) from it and from the momentum equation. Next, we insert the ansatz functions Eqs. (7) and (8) and postulate that the resulting equations hold for every \mathbf{r}' separately. For the momentum equation, one obtains

$$\frac{\partial}{\partial t} \mathbf{K}_{uu} + \mathbf{K}_{uu} \cdot \nabla \mathbf{u}_{\text{ref}} + \mathbf{u}_{\text{ref}} \cdot \nabla \mathbf{K}_{uu} = -\nabla \mathbf{K}_{pu} + \nu \nabla^2 \mathbf{K}_{uu} \quad (20)$$

or written componentwise

$$\frac{\partial}{\partial t} K_{ik}^{(uu)} + K_{jk}^{(uu)} u_{i,j}^{(\text{ref})} + u_j^{(\text{ref})} K_{ik,j}^{(uu)} = -K_{k,i}^{(pu)} + \nu K_{ik,jj}^{(uu)}, \quad (21)$$

where indices after a comma represent derivatives. It is important to note that the derivatives in Eq. (20) only act on the first coordinate of the propagators and that they are decoupled with regard to the second index. Hence, to obtain the tensorial \mathbf{K}_{uu} that depends on two spatial coordinates, one may solve a large number of vectorial equations in 3D.

The close connection of the propagators with the Green's functions

$$\mathbf{G}_{uu}(\mathbf{r}, \mathbf{r}', t) \equiv \theta(t) \mathbf{K}_{uu}(\mathbf{r}, \mathbf{r}', t), \quad (22)$$

$$\mathbf{G}_{pu}(\mathbf{r}, \mathbf{r}', t) \equiv \theta(t) \mathbf{K}_{pu}(\mathbf{r}, \mathbf{r}', t), \quad (23)$$

where $\theta(t)$ is the Heaviside function, can be easily confirmed on Eq. (20) using the initial condition Eq. (9). One finds

$$\frac{\partial}{\partial t} \mathbf{G}_{uu} + \mathbf{G}_{uu} \cdot \nabla \mathbf{u}_{\text{ref}} + \mathbf{u}_{\text{ref}} \cdot \nabla \mathbf{G}_{uu} + \nabla \mathbf{G}_{pu} - \nu \nabla^2 \mathbf{G}_{uu} = \delta(t) \delta(\mathbf{r} - \mathbf{r}') \mathbf{1}, \quad (24)$$

which is the very definition of a Green's function.

Having derived the propagator momentum equation, that for pressure takes the form

$$\text{Tr}[(\nabla \mathbf{K}_{uu}) \cdot (\nabla \mathbf{u}_{\text{ref}})] + \text{Tr}[(\nabla \mathbf{u}_{\text{ref}}) \cdot (\nabla \mathbf{K}_{uu})] = -\nabla^2 \mathbf{K}_{pu} \quad (25)$$

and

$$K_{jk,i}^{(uu)} u_{i,j}^{(\text{ref})} + K_{ik,j}^{(uu)} u_{j,i}^{(\text{ref})} = -K_{k,ii}^{(pu)}, \quad (26)$$

respectively. Ideally, a propagator obeying this equation leads to solenoidal velocity fields. However, numerical inaccuracies that accumulate over the course of many steps, e.g., because of discretization errors, could occur and might need to be corrected by solving a Poisson equation for the neglected contribution to pressure that restores solenoidality. As long as the effect remains small, convergence will be fast and the impact on the numerical costs not too severe.

The time evolution due to pressure variations is governed by similar equations with minor deviations. Although the continuity equation

$$\nabla \cdot \mathbf{K}_{up} = 0 \quad (27)$$

may be written down explicitly because the pressure field $\delta p(\mathbf{r}', 0)$ is not constrained as the velocity, the momentum equation,

$$\frac{\partial}{\partial t} \mathbf{K}_{up} + \mathbf{K}_{up} \cdot \nabla \mathbf{u}_{\text{ref}} + \mathbf{u}_{\text{ref}} \cdot \nabla \mathbf{K}_{up} = -\nabla \mathbf{K}_{pp} + \nu \nabla^2 \mathbf{K}_{up}, \quad (28)$$

$$\frac{\partial}{\partial t} K_i^{(up)} + K_j^{(up)} u_{i,j}^{(\text{ref})} + u_j^{(\text{ref})} K_{i,j}^{(up)} = -K_{,i}^{(pp)} + \nu K_{i,jj}^{(up)}, \quad (29)$$

and that for pressure,

$$\text{Tr}[(\nabla \mathbf{K}_{up}) \cdot (\nabla \mathbf{u}_{\text{ref}})] + \text{Tr}[(\nabla \mathbf{u}_{\text{ref}}) \cdot (\nabla \mathbf{K}_{up})] = -\nabla^2 \mathbf{K}_{pp}, \quad (30)$$

$$K_{j,i}^{(up)} u_{i,j}^{(\text{ref})} + K_{i,j}^{(up)} u_{j,i}^{(\text{ref})} = -K_{,ii}^{(pp)}, \quad (31)$$

are of identical structure as for the evolution due to velocity variations.

Unless $\mathbf{r}' \rightarrow \mathbf{r}_b \in \partial \Omega_{u,p}$, these equations are solved by $\mathbf{K}_{up}(\mathbf{r}, \mathbf{r}', t) = \mathbf{0}$ and $\mathbf{K}_{pp}(\mathbf{r}, \mathbf{r}', t) = 0$, which is a mere consequence from the fact that pressure inside the domain follows from the velocity field and can only be specified independently on the boundary.

If the pressure on $\partial \Omega_p$ is the same as that of the reference solution, then it is obvious that no contribution to the velocity and pressure prediction arises from $\partial \Omega_p$. Furthermore, for a pressure field that has the same gradient on $\partial \Omega_u$ as the reference state and hence a vanishing gradient of the deviation, these contributions of the propagator cancel each other out. This is a consequence from the following fact: If $\mathbf{K}_{up}(\mathbf{r}, \mathbf{r}_b \in \partial \Omega_u, t)$, $\mathbf{K}_{pp}(\mathbf{r}, \mathbf{r}_b \in \partial \Omega_u, t)$ solve the EOMs, then so do $-\mathbf{K}_{up}(\mathbf{r}, \mathbf{r}_b + \epsilon \mathbf{n}, t)$, $-\mathbf{K}_{pp}(\mathbf{r}, \mathbf{r}_b + \epsilon \mathbf{n}, t)$ for $\epsilon \rightarrow 0$ with an inwards pointing surface normal unit vector \mathbf{n} . It is straight forward to see that Eqs. (28) and (30) remain invariant under change of sign, and that the initial condition Eq. (10) and boundary conditions Eqs. (11) and (12) for \mathbf{K}_{up} hold. While the boundary condition Eq. (14) on $\partial \Omega_p$ for \mathbf{K}_{pp} evaluates to 0 and is therefore trivial if the second argument is located on $\partial \Omega_u$, that on $\partial \Omega_u$ given by Eq. (13) requires some thought. For each point \mathbf{r}_b on $\partial \Omega_u$, one has $\mathbf{K}_{pp}(\mathbf{r}_b + \epsilon \mathbf{n}, \mathbf{r}') - \mathbf{K}_{pp}(\mathbf{r}_b, \mathbf{r}') = \delta(\mathbf{r}_b + \epsilon \mathbf{n} - \mathbf{r}') - \delta(\mathbf{r}_b - \mathbf{r}')$. Only the cases $\mathbf{r}' = \mathbf{r}_b$ and $\mathbf{r}' = \mathbf{r}_b + \epsilon \mathbf{n}$ are relevant, for which one finds

$$\mathbf{K}_{pp}(\mathbf{r}_b + \epsilon \mathbf{n}, \mathbf{r}_b + \epsilon \mathbf{n}) - \mathbf{K}_{pp}(\mathbf{r}_b, \mathbf{r}_b + \epsilon \mathbf{n}) = \delta(0), \quad (32)$$

$$\mathbf{K}_{pp}(\mathbf{r}_b + \epsilon \mathbf{n}, \mathbf{r}_b) - \mathbf{K}_{pp}(\mathbf{r}_b, \mathbf{r}_b) = -\delta(0). \quad (33)$$

Therefore, $\mathbf{n} \cdot \nabla \mathbf{K}_{pp}(\mathbf{r}_b, \mathbf{r}_b) = -\mathbf{n} \cdot \nabla \mathbf{K}_{pp}(\mathbf{r}_b, \mathbf{r}_b + \epsilon \mathbf{n})$ on $\partial \Omega_u$, which proves the above statement.

To conclude, if neither the value of pressure on $\partial \Omega_p$ nor that of its gradient on $\partial \Omega_u$ differ from the reference solution p_{ref} , then it is not necessary to compute the evolution due to pressure deviations.

D. Strength of the propagators

We define the strength of the velocity-velocity propagator by its zeroth moment

$$\mathbf{k}_{uu}^{(0)}(\mathbf{r}, t) \equiv \int_{\Omega} d^3 r' \mathbf{K}_{uu}(\mathbf{r}, \mathbf{r}', t), \quad (34)$$

with analogous expressions for the other kernels. The propagator strengths of velocity deviations obey

$$\frac{\partial}{\partial t} \mathbf{k}_{uu}^{(0)} + \mathbf{k}_{uu}^{(0)} \cdot \nabla \mathbf{u}_{\text{ref}} + \mathbf{u}_{\text{ref}} \cdot \nabla \mathbf{k}_{uu}^{(0)} = -\nabla \mathbf{k}_{pu}^{(0)} + \nu \nabla^2 \mathbf{k}_{uu}^{(0)}, \quad (35)$$

$$\text{Tr}[(\nabla \mathbf{k}_{uu}^{(0)}) \cdot (\nabla \mathbf{u}_{\text{ref}})] + \text{Tr}[(\nabla \mathbf{u}_{\text{ref}}) \cdot (\nabla \mathbf{k}_{uu}^{(0)})] = -\nabla^2 \mathbf{k}_{pu}^{(0)}, \quad (36)$$

with initial condition

$$\mathbf{k}_{uu}^{(0)}(\mathbf{r}, t = 0) = \mathbf{1} \quad (37)$$

and boundary conditions

$$\left. \begin{aligned} \mathbf{k}_{uu}^{(0)}(\mathbf{r} \rightarrow \mathbf{r}_b) &= \mathbf{1} \\ \mathbf{n} \cdot \nabla \mathbf{k}_{pu}^{(0)}(\mathbf{r} \rightarrow \mathbf{r}_b) &= \mathbf{0}, \end{aligned} \right\} \text{ for } \mathbf{r}_b \in \partial \Omega_u \quad (38)$$

and

$$\left. \begin{aligned} \mathbf{n} \cdot \nabla \mathbf{k}_{uu}^{(0)}(\mathbf{r} \rightarrow \mathbf{r}_b) &= \mathbf{0} \\ \mathbf{k}_{pu}^{(0)}(\mathbf{r} \rightarrow \mathbf{r}_b) &= \mathbf{0} \end{aligned} \right\} \text{ for } \mathbf{r}_b \in \partial \Omega_p. \quad (39)$$

The integrated propagation due to pressure deviations can even be written down explicitly. Because of $\mathbf{k}_{up}^{(0)}(\mathbf{r}, t = 0) = \mathbf{0}$ and $\mathbf{k}_{pp}^{(0)}(\mathbf{r}_b \in \partial \Omega_p, t) = 1$, it follows in a straight-forward fashion that

$$\mathbf{k}_{up}^{(0)}(\mathbf{r}, t) = \mathbf{0}, \quad (40)$$

$$\mathbf{k}_{pp}^{(0)}(\mathbf{r}, t) = 1 \quad (41)$$

solve the volume-integrated versions of Eqs. (28) and (30).

Equations (35) and (36) are numerically rather tame with the initial condition being simply a constant instead of a sharp delta function, and Eqs. (40) and (41) even provide analytic solutions. Hence, they can be conveniently used to assess the accuracy of \mathbf{K}_{uu} , \mathbf{K}_{up} , \mathbf{K}_{pu} , and \mathbf{K}_{pp} obtained from the numerical solution of Eqs. (20), (25), (28), and (30) by computing their volume integrals and comparing the results with $\mathbf{k}_{uu}^{(0)}$, $\mathbf{k}_{up}^{(0)}$, $\mathbf{k}_{pu}^{(0)}$, and $\mathbf{k}_{pp}^{(0)}$.

III. CASE DESCRIPTION AND NUMERICAL SETUP

We implemented the deviation-propagation methodology in the open-source framework of OpenFOAM [52] in a standard finite-volume fashion. For the investigated case, no specific handling of the Dirac delta functions in the initial and boundary conditions, which became Kronecker deltas with appropriate prefactors upon discretization, was necessary. However, for high Reynolds number flows, a dedicated treatment might be to control numerical diffusion. The code and the case described below can be accessed in our repository [53].

In the present study, we applied our approach to the prototypical case of vortex shedding behind a cylinder of radius $r_{\text{cyl}} = 1$ m. For the sake of simplicity, we reduced to problem to two dimensions and studied a box with extensions $[-20 \text{ m}; 30 \text{ m}] \times [-20 \text{ m}; 20 \text{ m}]$, which was discretized into a mesh with 9200 cells. The Navier-Stokes equations were solved using the PISO algorithm with time steps of $\Delta t_{\text{CFD}} = 0.01$ s, which led to Courant numbers well below 1.0 for the investigated inflow velocities in the range from $u_{\text{in}} = 1.0$ m/s to $u_{\text{in}} = 1.2$ m/s. With a kinematic viscosity of $\nu = 0.01 \text{ m}^2/\text{s}$, these values corresponded to Reynolds numbers $\text{Re} = 200$ to $\text{Re} = 240$.

By monitoring drag and lift coefficients, we determined that data sampling of the reference time series with $u_{\text{in}} = 1.0$ m/s could start after an equilibration time of $t_{\text{equil}} = 180$ s. Due to the high degree of periodicity established after t_{equil} , we recorded flow fields for approximately one shedding cycle of $t_{\text{sample}} = 10$ s with a sampling step $\Delta t_{\text{sample}} = 0.01$ s equal to the solution step size.

Next, the propagators were obtained. To this end, the reference time series was split into ten intervals with durations $t_{\text{pred}} = 1$ s each, and for each of these segments, the corresponding \mathbf{K}_{uu} , \mathbf{K}_{pu} were calculated resulting in a database of ten reference states with an accompanying propagator each. Equations (20) and (25) were solved with time steps $\Delta t_{\text{prop}} = 0.0025$ s, 0.005 s, 0.01 s, and 0.02 s to assess their convergence behavior.

The deviation-propagation approach was evaluated by comparing its predictions with full-CFD data for the following case variations:

- (i) base case with uniform inlet velocity $u_{\text{in}} = 1.0$ m/s
- (ii) uniform inlet velocity $u_{\text{in}} = 1.1$ m/s
- (iii) uniform inlet velocity $u_{\text{in}} = 1.2$ m/s
- (iv) same inflow as base case, rotating cylinder with $\omega_{\text{cyl}} = \pi/4$ 1/s
- (v) same inflow as base case, rotating cylinder with $\omega_{\text{cyl}} = \pi/2$ 1/s
- (vi) same inflow as base case with superposed shear profile with $\partial u_x / \partial y = 0.2$ 1/s
- (vii) same inflow as base case with superposed shear profile with $\partial u_x / \partial y = 0.4$ 1/s

For cases (vi) and (vii), the shear profile was limited to $-1 \text{ m} \leq y \leq 1 \text{ m}$, resulting in a slanted step function. For all of the above scenarios, predictions for a single step t_{pred} and for multiple ones were made. In the latter case, after each step, the most similar reference state within the set of ten configurations was determined by evaluating Eq. (18) before the next prediction was made. The workflow is illustrated by Fig. 1. Since the propagators were constructed to keep the flow divergence-free, comparatively few iterations were sufficient to solve the equation for the remaining pressure and restore full solenoidality.

We stress that for all case variations, the same database of reference states corresponding to the base setup was used.

IV. RESULTS

After a brief exploration of the general properties of the velocity-velocity propagator, results on its prediction capabilities for specific settings of the prediction step and cutoff value are presented. Afterwards, the influence of these parameters is highlighted.

A. Propagators

A first impression of the structure of deviation propagators is provided by Fig. 2, where the Frobenius norm

$$|\mathbf{K}_{uu}|(\mathbf{r}, \mathbf{r}') \equiv \sqrt{\text{Tr}[\mathbf{K}_{uu}^T(\mathbf{r}, \mathbf{r}') \cdot \mathbf{K}_{uu}(\mathbf{r}, \mathbf{r}')] } \quad (42)$$

is displayed for two selected points \mathbf{r} . It can be seen how upstream points distributed around a small center of influence contributed to the propagator. With increasing distance from this center, the magnitude dropped off, which was a consequence of the diffusive term in the Navier-Stokes equations. For computational reasons, it was necessary to neglect any contributions $d^3 r' |\mathbf{K}_{uu}|(\mathbf{r}, \mathbf{r}') \leq K_{\text{min}}$ below a small, predefined threshold. Figure 3(a) demonstrates its impact in terms of the relative distance

$$D_1 \left[\mathbf{k}_{uu}^{(0)}, \int \mathbf{K}_{uu} \right] \equiv \frac{\int_{\Omega} d^3 r |\mathbf{k}_{uu}^{(0)}(\mathbf{r}) - \int_{\Omega} d^3 r' \mathbf{K}_{uu}(\mathbf{r}, \mathbf{r}')|}{\int_{\Omega} d^3 r |\mathbf{k}_{uu}^{(0)}(\mathbf{r})|} \quad (43)$$

between the integrated propagator $\int_{\Omega} d^3 r' \mathbf{K}_{uu}(\mathbf{r}, \mathbf{r}')$ and its strength $\mathbf{k}_{uu}^{(0)}(\mathbf{r})$ obtained from Eqs. (35) and (36). Below a value of about $K_{\text{min}}^{(\text{crit})} \approx 10^{-4}$, D_1 decreased much more rapidly with decreasing

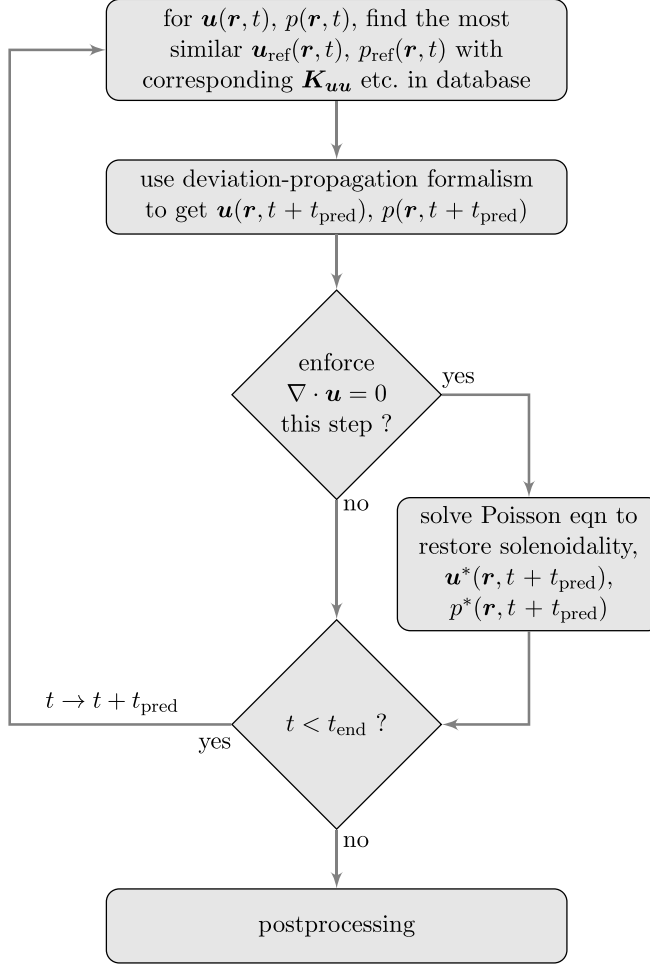
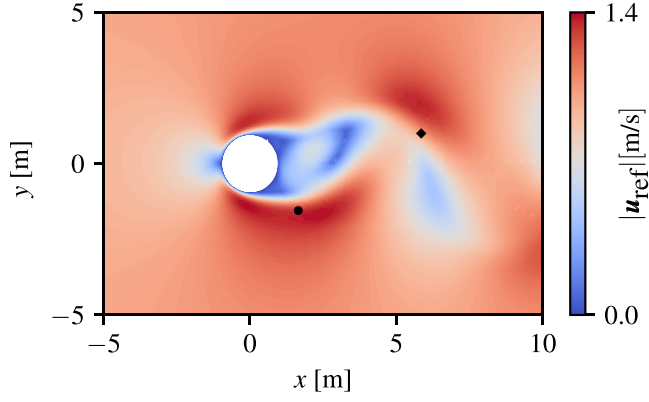


FIG. 1. Flowchart of deviation-propagation predictions. Once the most similar reference state has been identified, the accompanying propagators evolve the deviations by t_{pred} . If necessary, then lack of solenoidality due to numerical errors can be fixed before proceeding to the next step or to postprocessing.

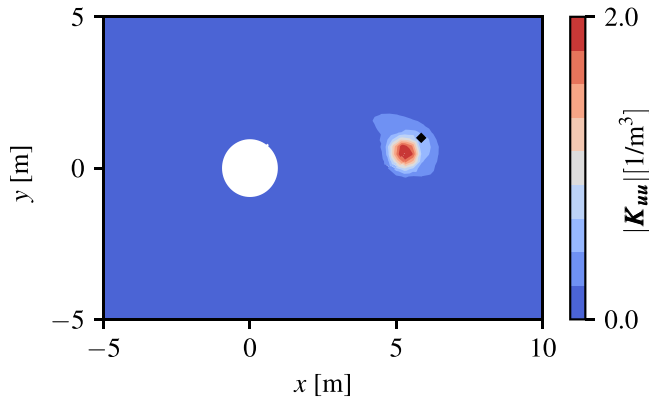
K_{\min} than above. This indicates that for larger values, the propagator was comparatively localized whereas for smaller ones, it was massively spread out. We hypothesized that the former contributions were the physically relevant ones and discarded the latter ones as a compromise between accuracy and memory demands. Aside from the fact that they stemmed at least partly from numerical diffusion, their effect for flow field predictions would have been very small. Their already minor contribution of about 1% to the overall strength of the propagator would have been further diminished because $\delta \mathbf{u}$ could change its sign repeatedly over larger regions so that these parts of the convolution would have canceled out.

As shown by Fig. 3(a), the integration time step Δt_{prop} had no impact on the strength of \mathbf{K}_{uu} . However, it took a sufficiently small step size for the spatially resolved propagators to converge. This behavior can be observed in Fig. 3(b), which shows the relative distance

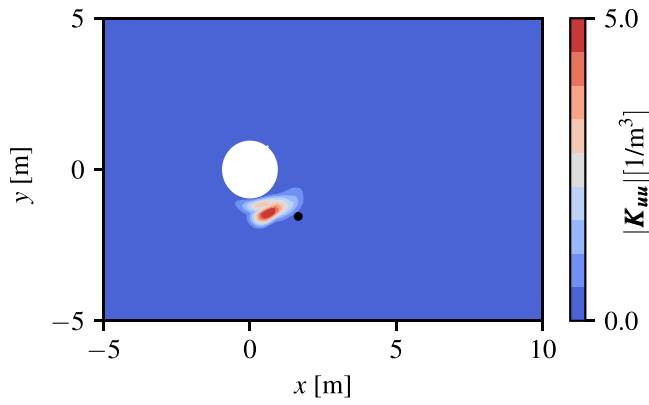
$$D_2[\mathbf{K}_{uu}^{(1)}, \mathbf{K}_{uu}^{(2)}] \equiv \frac{1}{V_\Omega} \int_\Omega d^3 r \frac{\int_\Omega d^3 r' |\mathbf{K}_{uu}^{(1)}(\mathbf{r}, \mathbf{r}') - \mathbf{K}_{uu}^{(2)}(\mathbf{r}, \mathbf{r}')|}{\sqrt{\int_\Omega d^3 r' |\mathbf{K}_{uu}^{(1)}(\mathbf{r}, \mathbf{r}')|} \sqrt{\int_\Omega d^3 r' |\mathbf{K}_{uu}^{(2)}(\mathbf{r}, \mathbf{r}')|}} \quad (44)$$



(a)



(b)



(c)

FIG. 2. Illustration of the flow and propagators. Panel (a) shows a snapshot of the velocity magnitude with two selected reference points (indicated by a diamond and a circle) for which the corresponding propagator norms according to Eq. (42) can be seen in panels (b) and (c).

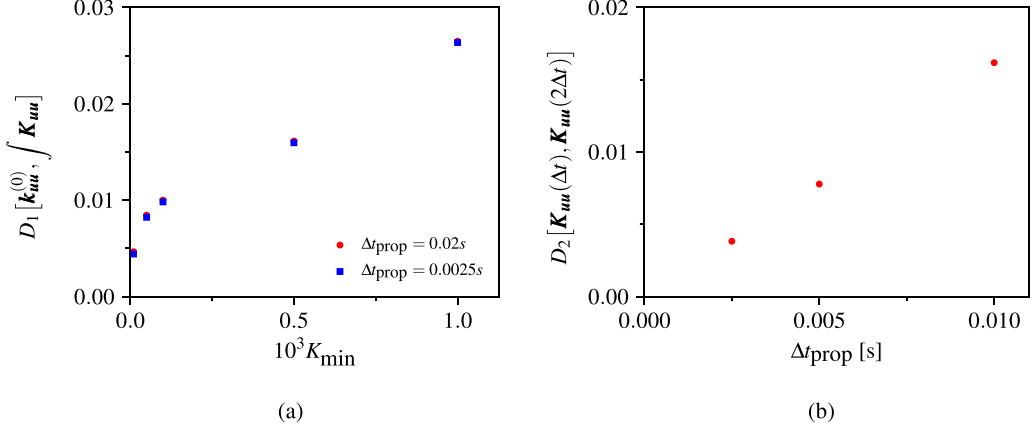


FIG. 3. Convergence of the velocity-velocity propagator. (a) The neglected strength of \mathbf{K}_{uu} decreased with decreasing K_{\min} . Below a critical value, the curve fell off rapidly. (b) The time step Δt_{prop} used for the calculation of the propagator with $K_{\min} = 10^{-4}$ needed to be sufficiently small to reach an acceptable degree of convergence. For both panels (a) and (b), each data point was averaged over all reference states.

between propagators computed with a time step Δt_{prop} and with twice its value. In the present case, using the same time step as for the underlying CFD calculations led to an error below 1%. Given the approximate nature of the deviation-propagation approach, such a value seems to be acceptable, but a larger step size cannot be recommended.

B. Predictions

To assess the quality of the deviation-propagation predictions, we first evaluated the relative error

$$E_{\text{pred}} \equiv \frac{\int_{\Omega} d^3r |\mathbf{u}_{\text{ex}}(\mathbf{r}, t + t_{\text{pred}}) - \mathbf{u}_{\text{pred}}(\mathbf{r}, t + t_{\text{pred}})|}{\int_{\Omega} d^3r |\mathbf{u}_{\text{ex}}(\mathbf{r}, t + t_{\text{pred}})|} \quad (45)$$

between the exact solution \mathbf{u}_{ex} and the prediction \mathbf{u}_{pred} made by Eqs. (7) after a single step. For each of the case variations described in Sec. III, we used 61 flow fields drawn from a time series with sampling step 0.5s and predicted the evolution of each of them. Figure 4(a) shows a clear correlation with the error E_{lin} which was made in linearizing the Navier-Stokes equations. This demonstrates its capability to identify nearest neighbors. The more similar a reference state from the prerecorded database could be found to a flow field for which a prediction should be made, the smaller the resulting error was. Notably, predictions with a step size of $t_{\text{pred}} = 100\Delta t_{\text{CFD}}$ were remarkably accurate. Even for the demanding case of a rotating cylinder, the relative error was below 1%.

It had to be expected that the error after multiple consecutive steps had accumulated. It can be seen in Fig. 4(b) that it grew the fastest for those cases where the single step predictions had been the most inaccurate. While for the base case (same boundary conditions but an initial condition not present among the reference states), hardly any deviation from the exact solution had established even after 20 steps corresponding to 20s or approximately two shedding cycles, the case with the rotating cylinder showed a relative error of about 5%. However, the growth of the error had slowed down, and a similar behavior could be found for several of the other cases: a linear regime was followed by a slower increase. Even though the prediction accuracy was not extremely good anymore after this relatively long evolution time (compared with the underlying CFD time step), it seemed that long-term predictions with a much larger number of steps might be possible.

This hope is indeed confirmed by Figs. 5 and 6, which show the time-averaged velocity fields for some of the investigated cases after 100s corresponding to approximately ten shedding cycles.

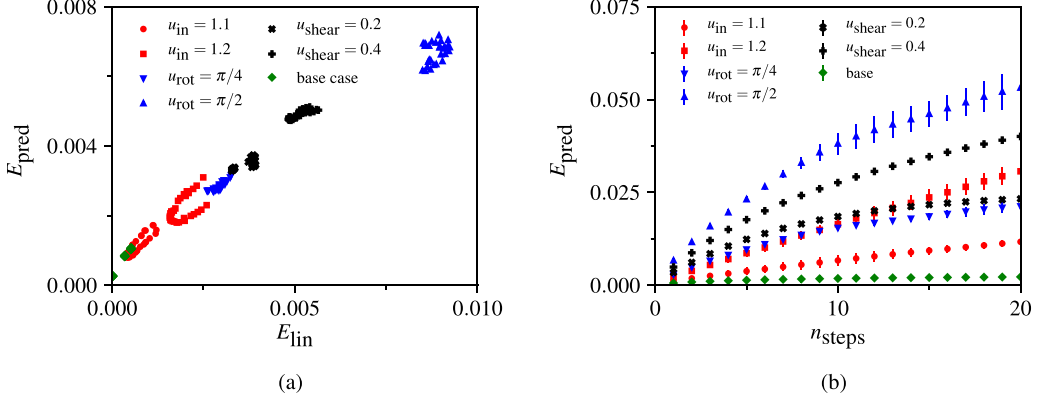


FIG. 4. Single and multi step prediction errors. (a) After a single prediction step, errors for all cases were below 1%, depending on the magnitude of E_{lin} . (b) The errors increased with several, consecutive steps. Error bars indicate a certain degree of scattering obtained from 61 time series with different initial states.

No deviation is visible at all for the base case in Figs. 5(a) and 5(b), and hardly can any differences can be found for the case with higher, uniform inlet velocity $u_{\text{in}} = 1.2$ m/s in Figs. 5(c) and 5(d). A cylinder rotation of $u_{\text{rot}} = \pi/2$ 1/s and a shear profile with $\partial u_x / \partial y = 0.4$ 1/s led to minor, quantitative differences that can be seen in Figs. 6(a)–6(d), respectively. However, the qualitative agreement was still very good considering that the database upon which predictions were made had

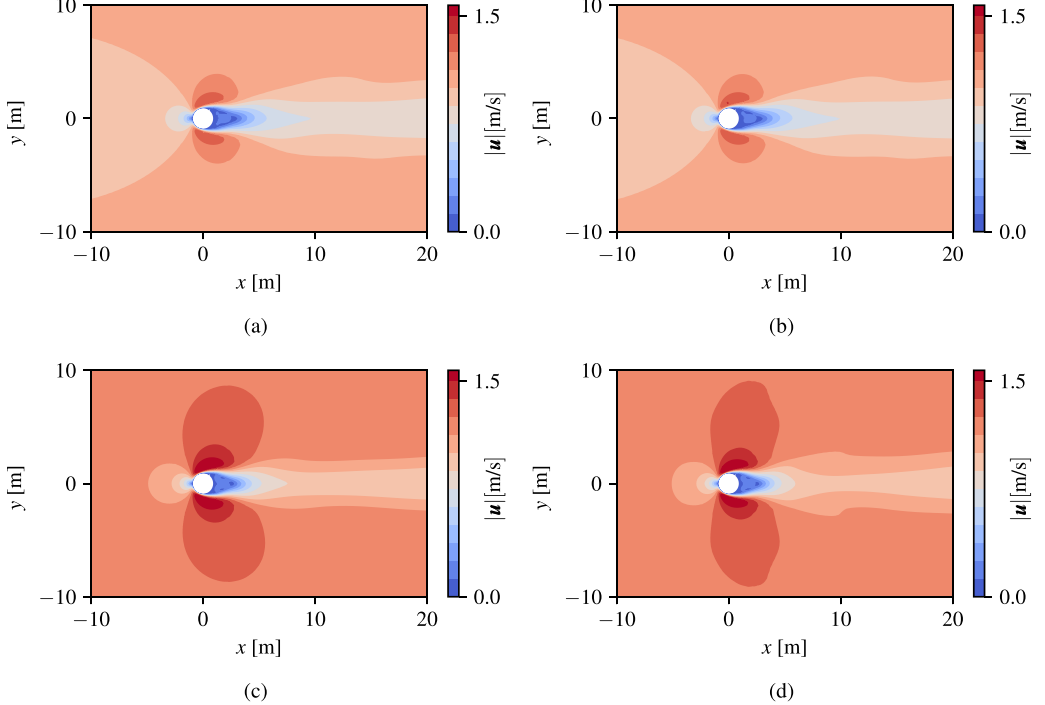


FIG. 5. Time-averaged flow fields. There were hardly any deviations for the base case (upper row) between full CFD simulations (left column) and deviation-propagation predictions (right column). A similar level of agreement was reached for the case of uniform inlet velocity $u_{\text{in}} = 1.2$ m/s (lower row).

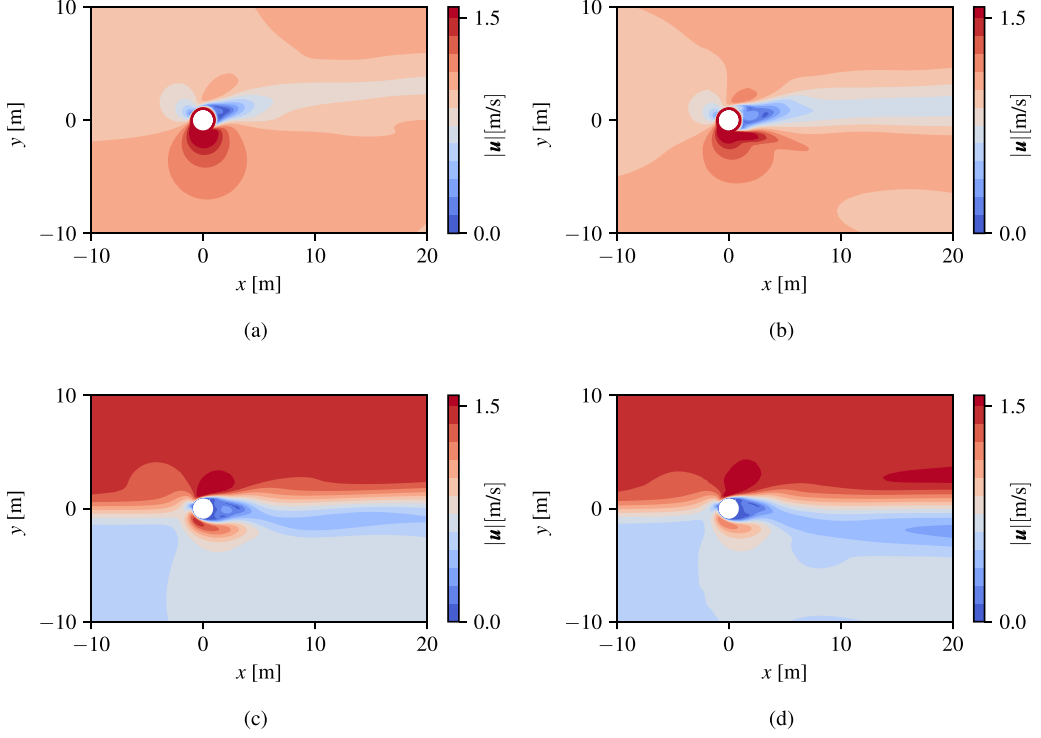


FIG. 6. Time-averaged flow fields under more demanding conditions. While local features of the exact solution (left column) differed from those predicted by the deviation-propagation formalism (right column), there were clear, qualitative similarities. Both a cylinder spinning with $u_{\text{rot}} = \pi/2$ 1/s (upper row) and a shear inlet profile with $\partial u_x / \partial y = 0.4$ 1/s (lower row) led to a deflection of the vortex street that was comparable for the full CFD and the deviation-propagation simulation.

the temporal average provided in Fig. 5(a). In particular, it is noteworthy that the velocity fields close to the cylinder wall had very low values in the reference time series, whereas there were significantly larger velocities when the cylinder rotated or a shear profile was applied. Such conditions clearly challenged the validity of our linearization approach, which makes the prediction quality even more remarkable.

A more quantitative assessment of the prediction accuracy is provided by the relative errors along the lines of Eq. (45) for the time averages in Table I. Even the very demanding cases of high rotation

TABLE I. Relative error of time-averaged predictions.

Case	Rel. error
Base	0.00065
$u_{\text{in}} = 1.1$ m/s	0.0041
$u_{\text{in}} = 1.2$ m/s	0.0083
$u_{\text{rot}} = \pi/4$ 1/s	0.014
$u_{\text{rot}} = \pi/2$ 1/s	0.035
$\partial u_x / \partial y = 0.2$ 1/s	0.013
$\partial u_x / \partial y = 0.4$ 1/s	0.028

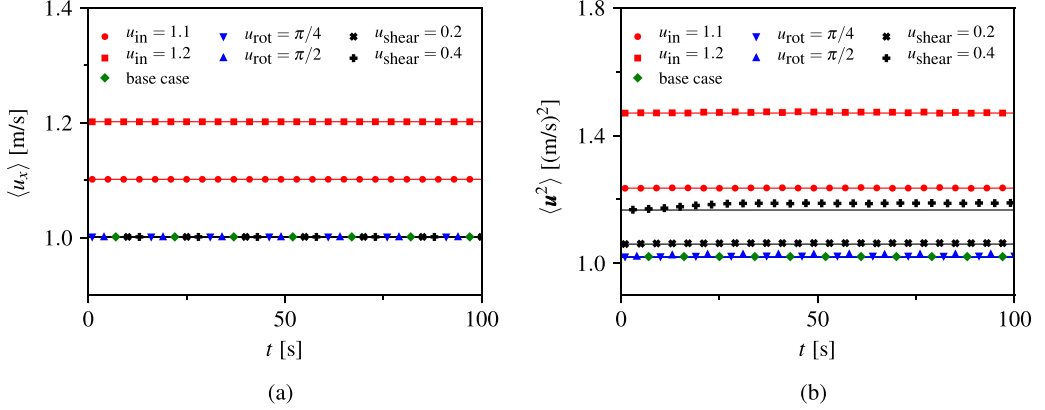


FIG. 7. Spatially averaged velocity and squared velocity. Straight lines resemble the results from full CFD calculations and symbols those from deviation-propagation predictions. No errors are visible regarding (a) the average velocity, and only a minor deviation for (b) the squared velocity of the shear-inlet profile can be seen.

rate and shear rate only caused relative errors of 3.5% and 2.8%. The investigated uniform variations of the inlet velocity, however, led to deviations below 1% and the base case had an utterly negligible error.

Besides the very good agreement of the long-term time averages of the predicted fields with the exact solution, it is also instructive to study the temporal evolution of global flow properties such as the volume-averaged velocity and squared velocity

$$\langle \mathbf{u} \rangle(t) = \frac{1}{V_\Omega} \int_\Omega d^3r \mathbf{u}(\mathbf{r}, t), \quad (46)$$

$$\langle \mathbf{u}^2 \rangle(t) = \frac{1}{V_\Omega} \int_\Omega d^3r \mathbf{u}^2(\mathbf{r}, t). \quad (47)$$

In Fig. 7, one observes an almost perfect conservation behavior with only one minor exception. The case with the higher shear rate had a slightly too high kinetic energy while all other cases fitted extremely well to the results obtained from a full CFD calculation.

C. Influence of propagator cutoff and prediction step size

It is instructive to understand how the quality of the results presented in the previous section depends on the most important parameters of the propagators. As already discussed, it was necessary to impose a cutoff K_{\min} below which contributions of the propagators were neglected. In the present case, we found $K_{\min} = 10^{-4}$ a reasonable choice. Figure 8 demonstrates how a larger value impaired the prediction accuracy. It is not surprising that an extreme choice such as $K_{\min} = 10^{-2}$, where a significant portion of the propagator strength was missing, could cause much larger errors. Especially the cases of increased uniform and shear inflow velocity were strongly affected, while the others showed less severely increased errors. This observation may be explained by the fact that deviations for the former two types of flow would change their sign less rapidly as for a spinning cylinder. Hence, their contributions did not cancel out as easily.

With $K_{\min} = 10^{-3}$, results were of similar quality as the reference with $K_{\min} = 10^{-4}$ except from those with increased uniform inlet velocity, where the effect mentioned above was the most pronounced. Since the same propagators were used for all case variants, it follows that a reasonable cutoff value depends on the target conditions to be simulated and not only on \mathbf{K} itself. For many cases, a generous choice of K_{\min} can suffice while certain flow configurations require to retain

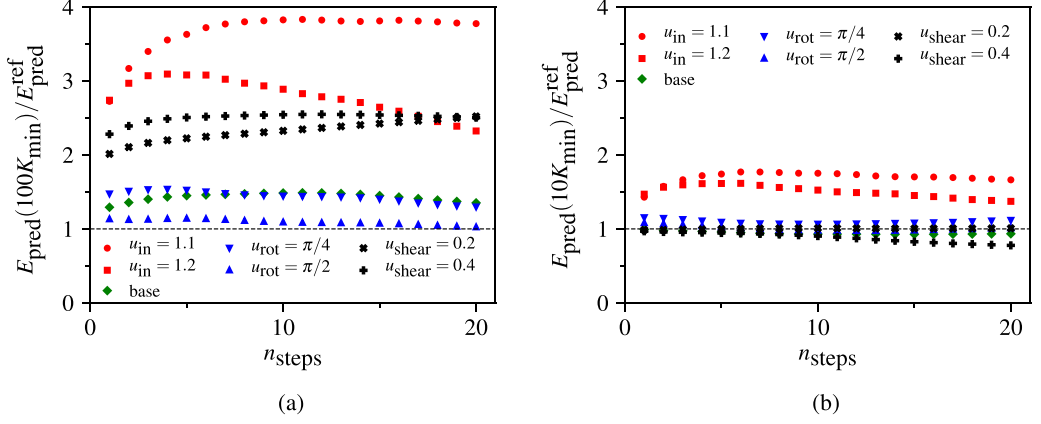


FIG. 8. Errors for different propagator cutoff values. Compared to predictions with a reference value of $K_{\text{min}} = 10^{-4}$, those made with (a) $100K_{\text{min}}$ were significantly larger for the cases with increased uniform and shear inflow. The base case and those with a rotating cylinder were less affected. (b) For $10K_{\text{min}}$, errors were generally smaller, and only the configurations with increased uniform inflow showed significant deviations.

contributions down to $K_{\text{min}}^{(\text{crit})}$. To reduce memory demands and computational costs, it is advisable to use as large a cutoff value as acceptable with regard to prediction accuracy.

Besides the spatial range over which the convolution operations in Eqs. (7) and (8) were carried out, the prediction step size influenced the quality of the obtained results. However, its effect was relatively weak. It can be seen in Fig. 9 that making one prediction with $2t_{\text{pred}}$ was only slightly more inaccurate than two with t_{pred} . In the beginning, mainly the cases with increased uniform inlet velocity were affected. We believe a similar mechanism as discussed above was at work. With a larger prediction step size, the propagators became more extended, and the cutoff K_{min} led to slightly larger errors. After multiple steps, however, most case variants showed errors of similar magnitude. Besides the issue of decreasing propagator localization, the less frequent identification of the nearest neighbor state in the database might have had some impact. Due to the larger steps covered by the integral kernels, it took longer until deviations were computed from a different time series segment with a higher degree of similarity.

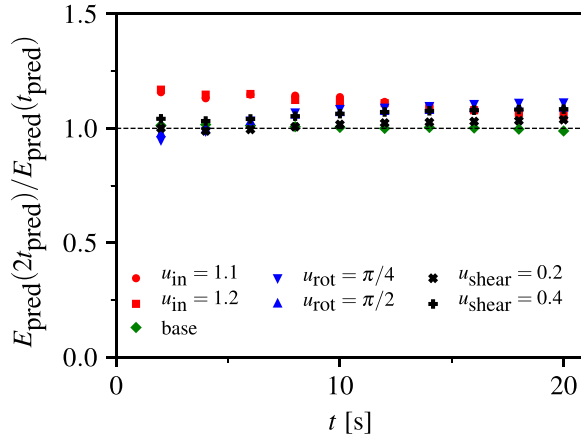


FIG. 9. Errors for different prediction steps. Doubling the step size to $2t_{\text{pred}}$ increased the prediction errors only moderately compared to those with t_{pred} .

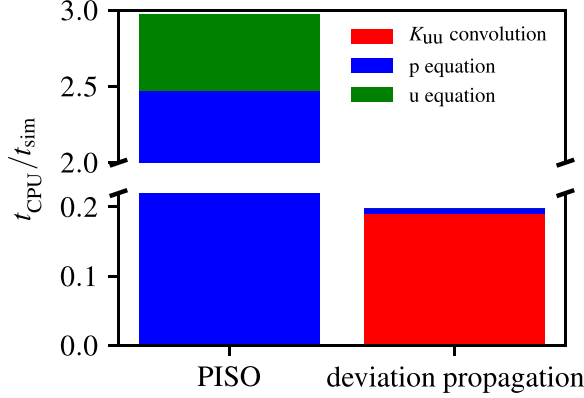


FIG. 10. CPU time per unit of simulated process time. For the PISO algorithm, the total numerical costs consisted predominantly of those for solving the pressure and the momentum equations. In the deviation-propagation simulations, most time was spent evaluating the convolution integrals and only a small fraction to restore solenoidality. The identification of nearest-neighbor states was negligible in the present case.

The above considerations demonstrate the usefulness of convergence and parameter variation studies for deviation-propagation simulations. Besides the issue of a reasonable compromise for K_{\min} , a sensible choice for t_{pred} needs to be made. Too large a value would lead to inaccurate predictions (together with massive memory demands for each propagator) whereas too small a value might not improve accuracy as much as desired, require a large number of reference states with accompanying propagators and impair the execution time. While they need not coincide, the temporal spacing between reference states [to be obtained with the aid of Eq. (17)] seems to be a plausible first guess for the propagator step size.

D. Performance analysis

Although the present investigation has mainly demonstrated that a linearized propagator approach can approximate simple flows reasonable well, but did not attempt any optimization, a comparison with the runtime of the PISO algorithm is nevertheless instructive. Figure 10 reproduces the well-known fact that most of the computational effort of PISO-based simulations was due to the solution of the Poisson equation for pressure, while the momentum equation accounted for a significantly smaller fraction of CPU time. Remarkably, the numerical costs for deviation-propagation calculations were distributed in an opposite fashion. Almost all time was spent with the explicit step of computing the convolution integrals for large-step predictions, whereas the implicit pressure correction required very little effort. Compared to its counterpart in the PISO algorithm, it had to be solved much less frequently due to the larger time step size, and it took fewer iterations per step because the propagators already accounted for incompressibility. The pressure correction mainly had to remove compressibility effects due to numerical inaccuracies.

Overall, the deviation-propagation simulations with a time step of $t_{\text{pred}} = 100\Delta t_{\text{CFD}}$ ran about 15 times faster than the PISO calculations with standard settings. The former were about five times as fast as the simulated process, while the latter were three times slower. We did not attempt to explore the highest possible speed up at this point, but stress the very different contributions to runtime for both methods. Especially the fact that deviation-propagation calculations spent almost all CPU time in the explicit prediction step will allow for significant improvements in future work in terms of parallelization and a more efficient evaluation of the convolution integrals.

Even though a further, significant acceleration of the approach seems to be within reach, it is legitimate to ask if similar speed ups to those observed in this study might be achieved by simply decreasing the target accuracy of standard CFD algorithms. In Figure 11, the deviation Eq. (45) of

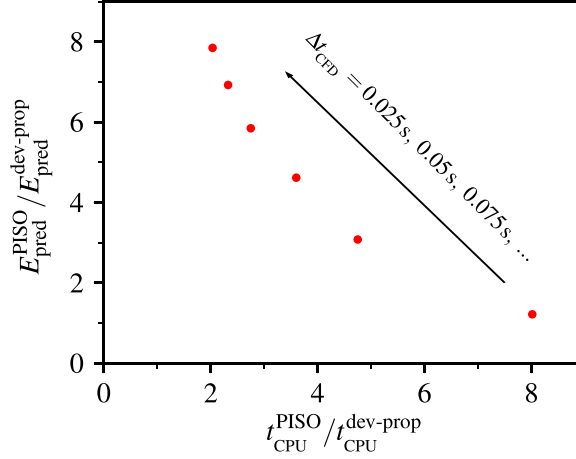


FIG. 11. CPU times and prediction errors for the PISO algorithm with different time step sizes. CPU times and errors are provided relative to those of deviation-propagation simulations for the base case. The reference velocity field towards which the errors were computed was obtained with $\Delta t_{\text{CFD}} = 0.01$ s.

the long-term averaged velocity field obtained with the PISO algorithm using increasing time step sizes is shown together with the required CPU time for a fixed process duration. With sufficiently large steps, the runtime decreased towards that of deviation-propagation simulations. At the same time, the overall error increased strongly. At twice the CPU time, it was approximately eight times larger than that in the deviation-propagation calculations for the base case. This shows that a similar speed up cannot be bought easily for conditions present in the database. Of course, off-database simulations come with larger prediction errors where such a comparison would be more favorable for standard CFD algorithms, especially ones of higher order [37–39], with reduced accuracy requirements. Ultimately, the adequacy of the propagator approach will depend on how far away from the attractor in the database one wants to carry out simulations and how much one can further speed up the evaluation of the convolution integrals.

V. CONCLUSION AND FUTURE WORK

This work has introduced a data-assisted and physics-informed combination of the method of analogs with a propagator approach for fluid mechanical problems. Given an initial flow state, a similar reference configuration is identified in a prerecorded database. Since its temporal evolution is known and that of the deviation between the two can be obtained employing their propagator, predictions with very large steps are possible. Put differently, the nonlinearity of the full Navier-Stokes equations is taken care of by the data-driven method of analogs while deviations from the reference states are governed by a linearized version of the Navier-Stokes equations, for which a large-step propagator can be computed.

For a number of case variations of vortex shedding, step sizes of up to 200 times larger than that used to generate the underlying data led to notably accurate results considering the magnitude of the prediction steps. With regard to the practical applicability of the method, its long-term behavior was of specific interest. The obtained temporal averages after approximately ten shedding cycles were very close to full CFD results, and excellent conservation of global properties such as total momentum and kinetic energy was found. No divergence issues due to the potential accumulation of prediction errors arose.

The main purpose of our study was the introduction of an alternative simulation technique on a conceptual level. For this reason, we restricted ourselves to a small-scale test case. Due to the low number of cells, which is not necessarily representative for more demanding scenarios, we carried

out only a rather basic performance analysis and evaluation of the numerical costs without going into intricate details. However, already these simple considerations demonstrated the explicit nature of the deviation-propagation formalism in contrast to many classical CFD algorithms and its potential for further improvements.

Before more demanding applications such as the study of temporal multiscale problems of highly dynamic flows, their control or parameter variation and optimization involving a large number of simulations can be targeted, several steps will have to be taken. In particular, two issues are very pressing: (i) The computation of the propagators needs to be massively accelerated, and (ii) their representation has to become more efficient to reduce the tremendous memory demands. Both problems could be cured with either an approximation of the propagators in terms of a few leading-order moments such as Eq. (34) or a clever choice of ansatz function with a few optimization parameters. As long as propagators are localized to a region of influence, they might be approximated, e.g., as (a sum of) Gaussians. Hence, for each point in the domain, one would only have to determine and store the corresponding small set of characteristic parameters. If a more elaborate treatment turns out to be necessary, then PINNs would offer greater flexibility to approximate the propagators.

Once the handling of the deviation-propagation approach has become more efficient, it will be interesting to apply it to more demanding conditions such as multiscale and/or multiphase flows, where coupled propagator equations and/or higher-order terms beyond a first-order expansion need to be dealt with.

-
- [1] S. Brunton, B. Noack, and P. Koumoutsakos, Machine learning for fluid mechanics, [Annu. Rev. Fluid Mech.](#) **52**, 477 (2020).
 - [2] M. Cheng, F. Fang, C. C. Pain, and I. Navon, Data-driven modelling of nonlinear spatio-temporal fluid flows using a deep convolutional generative adversarial network, [Comput. Methods Appl. Mech. Eng.](#) **365**, 113000 (2020).
 - [3] N. Akkari, F. Casenave, M.-E. Perrin, and D. Ryckelynck, Deep convolutional generative adversarial networks applied to 2D incompressible and unsteady fluid flows, in *Intelligent Computing* (Springer, Berlin, 2020), pp. 264–276.
 - [4] C. Quilodrán-Casas and R. Arcucci, A data-driven adversarial machine learning for 3D surrogates of unstructured computational fluid dynamic simulations, [Physica A](#) **615**, 128564 (2023).
 - [5] Z. Deng, C. He, Y. Liu, and K. C. Kim, Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework, [Phys. Fluids](#) **31**, 125111 (2019).
 - [6] N. Janssens, M. Huysmans, and R. Swennen, Computed tomography 3D super-resolution with generative adversarial neural networks: Implications on unsaturated and two-phase fluid flow, [Materials](#) **13**, 1397 (2020).
 - [7] M. Z. Yousif, L. Yu, and H.-C. Lim, Super-resolution reconstruction of turbulent flow fields at various Reynolds numbers based on generative adversarial networks, [Phys. Fluids](#) **34**, 015130 (2022).
 - [8] M. Bergmann, C.-H. Bruneau, and A. Iollo, Enablers for robust POD models, [J. Comput. Phys.](#) **228**, 516 (2009).
 - [9] S. Lorenzi, A. Cammi, L. Luzzi, and G. Rozza, POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations, [Comput. Methods Appl. Mech. Eng.](#) **311**, 151 (2016).
 - [10] G. Stabile, S. Hijazi, A. Mola, S. Lorenzi, and G. Rozza, POD-Galerkin reduced order methods for CFD using finite volume discretisation: Vortex shedding around a circular cylinder, [Commun. Appl. Ind. Math.](#) **8**, 210 (2017).
 - [11] A. V. G. Cavalieri and P. A. S. Nogueira, Reduced-order Galerkin models of plane Couette flow, [Phys. Rev. Fluids](#) **7**, L102601 (2022).

- [12] L. Agostini, Exploration and prediction of fluid dynamical systems using auto-encoder technology, *Phys. Fluids* **32**, 067103 (2020).
- [13] K. Fukami, T. Nakamura, and K. Fukagata, Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data, *Phys. Fluids* **32**, 095110 (2020).
- [14] T. Murata, K. Fukami, and K. Fukagata, Nonlinear mode decomposition with convolutional neural networks for fluid dynamics, *J. Fluid Mech.* **882**, A13 (2020).
- [15] K. Fukami, T. Murata, K. Zhang, and K. Fukagata, Sparse identification of nonlinear dynamics with low-dimensionalized flow representations, *J. Fluid Mech.* **926**, A10 (2021).
- [16] J. Page, M. P. Brenner, and R. R. Kerswell, Revealing the state space of turbulence using machine learning, *Phys. Rev. Fluids* **6**, 034402 (2021).
- [17] C. E. P. De Jesús and M. D. Graham, Data-driven low-dimensional dynamic model of Kolmogorov flow, *Phys. Rev. Fluids* **8**, 044402 (2023).
- [18] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proc. Natl. Acad. Sci. USA* **113**, 3932 (2016).
- [19] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, Data-driven discovery of coordinates and governing equations, *Proc. Natl. Acad. Sci. USA* **116**, 22445 (2019).
- [20] J.-C. Loiseau, Data-driven modeling of the chaotic thermal convection in an annular thermosyphon, *Theor. Comput. Fluid Dyn.* **34**, 339 (2020).
- [21] Y. Guan, S. L. Brunton, and I. Novosselov, Sparse nonlinear models of chaotic electroconvection, *R. Soc. Open Sci.* **8**, 202367 (2021).
- [22] M. Raissi and G. E. Karniadakis, Hidden physics models: Machine learning of nonlinear partial differential equations, *J. Comput. Phys.* **357**, 125 (2018).
- [23] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* **378**, 686 (2019).
- [24] M. Raissi, A. Yazdani, and G. E. Karniadakis, Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations, *Science* **367**, 1026 (2020).
- [25] M. Raissi, H. Babaei, and P. Givi, Deep learning of turbulent scalar mixing, *Phys. Rev. Fluids* **4**, 124501 (2019).
- [26] X. I. A. Yang, S. Zafar, J.-X. Wang, and H. Xiao, Predictive large-eddy-simulation wall modeling via physics-informed neural networks, *Phys. Rev. Fluids* **4**, 034602 (2019).
- [27] F. Sahli Costabal, Y. Yang, P. Perdikaris, D. E. Hurtado, and E. Kuhl, Physics-informed neural networks for cardiac activation mapping, *Front. Phys.* **8**, 42 (2020).
- [28] S. Cai, Z. Wang, S. Wang, P. Perdikaris, and G. E. Karniadakis, Physics-informed neural networks for heat transfer problems, *J. Heat Transfer* **143**, 060801 (2021).
- [29] B. Reyes, A. A. Howard, P. Perdikaris, and A. M. Tartakovsky, Learning unknown physics of non-Newtonian fluids, *Phys. Rev. Fluids* **6**, 073301 (2021).
- [30] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, Physics-informed machine learning, *Nat. Rev. Phys.* **3**, 422 (2021).
- [31] T. Lichtenegger, E. A. J. F. Peters, J. A. M. Kuipers, and S. Pirker, A recurrence CFD study of heat transfer in a fluidized bed, *Chem. Eng. Sci.* **172**, 310 (2017).
- [32] S. Abbasi, S. Puttering, S. Pirker, and T. Lichtenegger, Recurrence analysis and time extrapolation of a confined turbulent jet using modal decomposition, *Phys. Fluids* **32**, 075115 (2020).
- [33] F. Cecconi, M. Cencini, M. Falcioni, and A. Vulpiani, Predicting the future from the past: An old problem from a modern perspective, *Am. J. Phys.* **80**, 1001 (2012).
- [34] S. Pirker and T. Lichtenegger, Efficient time-extrapolation of single- and multiphase simulations by transport based recurrence CFD (rCFD), *Chem. Eng. Sci.* **188**, 65 (2018).
- [35] T. Lichtenegger, S. Abbasi, and S. Pirker, Transport in turbulent, recurrent flows: Time-extrapolation and statistical symmetrization, *Chem. Eng. Sci.* **259**, 117795 (2022).
- [36] S. Wang, X. Yu, and P. Perdikaris, When and why PINNs fail to train: A neural tangent kernel perspective, *J. Comput. Phys.* **449**, 110768 (2022).

- [37] H. Bijl, M. H. Carpenter, V. N. Vatsa, and C. A. Kennedy, Implicit time integration schemes for the unsteady compressible Navier–Stokes equations: Laminar flow, *J. Comput. Phys.* **179**, 313 (2002).
- [38] D. Forti and L. Dedè, Semi-implicit BDF time discretization of the Navier-Stokes equations with VMS-LES modeling in a high performance computing framework, *Comput. Fluids* **117**, 168 (2015).
- [39] K. C. Loy and Y. Bourgault, Robust high-order semi-implicit backward difference formulas for Navier-Stokes equations, *Int. J. Numer. Methods Fluids* **90**, 195 (2019).
- [40] D. Cacuci, R. Perez, and V. Protopopescu, Duals and propagators: A canonical formalism for nonlinear equations, *J. Math. Phys.* **29**, 353 (1988).
- [41] D. Cacuci and V. Protopopescu, Propagators for nonlinear systems, *J. Phys. A: Math. Gen.* **22**, 2399 (1989).
- [42] C. R. Gin, D. E. Shea, S. L. Brunton, and J. N. Kutz, DeepGreen: deep learning of Green’s functions for nonlinear boundary value problems, *Sci. Rep.* **11**, 21614 (2021).
- [43] N. Boullé, C. J. Earls, and A. Townsend, Data-driven discovery of Green’s functions with human-understandable deep learning, *Sci. Rep.* **12**, 4824 (2022).
- [44] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, *Nat. Mach. Intell.* **3**, 218 (2021).
- [45] Z. Li, H. Zheng, N. Kovachki, D. Jin, H. Chen, B. Liu, K. Azizzadenesheli, and A. Anandkumar, Physics-informed neural operator for learning partial differential equations, [arXiv:2111.03794](https://arxiv.org/abs/2111.03794).
- [46] S. Wang, H. Wang, and P. Perdikaris, Learning the solution operator of parametric partial differential equations with physics-informed DeepONets, *Sci. Adv.* **7**, eabi8605 (2021).
- [47] S. Wang and P. Perdikaris, Long-time integration of parametric evolution equations with physics-informed DeepONets, *J. Comput. Phys.* **475**, 111855 (2023).
- [48] K. Gustafson and R. Hartman, Divergence-free bases for finite element schemes in hydrodynamics, *SIAM J. Numer. Anal.* **20**, 697 (1983).
- [49] X. Ye and C. A. Hall, A discrete divergence-free basis for finite element methods, *Numer. Algorithms* **16**, 365 (1997).
- [50] P. W. Schroeder and G. Lube, Pressure-robust analysis of divergence-free and conforming FEM for evolutionary incompressible Navier–Stokes flows, *J. Numer. Math.* **25**, 249 (2017).
- [51] A. Allendes, G. R. Barrenechea, and J. Novo, A divergence-free stabilized finite element method for the evolutionary Navier–Stokes equations, *SIAM J. Sci. Comput.* **43**, A3809 (2021).
- [52] OpenFOAM, <https://openfoam.org/>
- [53] <https://github.com/ParticulateFlow/pfmFOAM-public>