Compressive sensing based machine learning strategy for characterizing the flow around a cylinder with limited pressure measurements

Ido Bright, Guang Lin, and J. Nathan Kutz

Citation: Physics of Fluids (1994-present) **25**, 127102 (2013); doi: 10.1063/1.4836815
View online: http://dx.doi.org/10.1063/1.4836815
View Table of Contents: http://scitation.aip.org/content/aip/journal/pof2/25/12?ver=pdfcov
Published by the AIP Publishing

PHYSICS OF FLUIDS **25**, 127102 (2013)

Compressive sensing based machine learning strategy for characterizing the flow around a cylinder with limited pressure measurements

Ido Bright,[1] Guang Lin,[2] and J. Nathan Kutz[1,a)]
[1]*Department of Applied Mathematics, University of Washington, Seattle, Washington 98195-2420, USA*
[2]*Pacific Northwest National Laboratory, PO Box 999, Richland, Washington 99352, USA*

(Received 22 April 2013; accepted 15 November 2013; published online 5 December 2013)

Compressive sensing is used to determine the flow characteristics around a cylinder (Reynolds number and pressure/flow field) from a sparse number of pressure measurements on the cylinder. Using a supervised machine learning strategy, library elements encoding the dimensionally reduced dynamics are computed for various Reynolds numbers. Convex L_1 optimization is then used with a limited number of pressure measurements on the cylinder to reconstruct, or decode, the full pressure field and the resulting flow field around the cylinder. Aside from the highly turbulent regime (large Reynolds number) where only the Reynolds number can be identified, accurate reconstruction of the pressure field and Reynolds number is achieved. The proposed data-driven strategy thus achieves encoding of the fluid dynamics using the L_2 norm, and robust decoding (flow field reconstruction) using the sparsity promoting L_1 norm. © *2013 AIP Publishing LLC*. [http://dx.doi.org/10.1063/1.4836815]

I. INTRODUCTION

Time-varying fluid flows are ubiquitous in modern engineering and in the life sciences. Particularly challenging is the characterization of unsteady aerodynamic forces and moments as they play critical roles in, for instance, biological propulsion and bio-inspired engineering design principals.[1–4] It is observed that birds, bats, insects, and fish routinely harness unsteady fluid phenomena to improve their propulsive efficiency, maximize thrust and lift, and increase maneuverability.[5–10] And despite their limited number of computational resources and spatially distributed, noisy sensors, they achieve robust flight dynamics and control despite potentially large perturbations in flight conditions (e.g., wind gusts, bodily harm). Such observations are highly suggestive, leading to conjectures about the existence of low-dimensional structures in the flow-field that are sparsely sampled and which are exploited for robust control purposes. Our goal is to develop a mathematical framework for directly addressing this conjecture. In particular, we consider the model problem of flow around a cylinder where (i) dimensionality reduction is used to *encode* the low-dimensional dynamics present in the system for a wide range of fluid flows, and (ii) compressive (sparse) sampling is used to robustly *decode* the dynamic state of the system, thus revealing a promising, low-dimensional control architecture. The integration of such encoding (machine learning) and decoding (compressive sensing) strategies represents a novel approach for characterizing complex dynamical systems and/or fluid flows. Indeed, this approach presents a tremendous opportunity to develop the underlying mathematical theory of dynamical systems in the context of sparse data assimilation and machine learning.

Like many fields, fluid dynamics is experiencing a tremendous boon in the development of experimental and computational tools capable of generating enormous amounts of data, thus fueling

a)Author to whom correspondence should be addressed. Electronic mail: kutz@uw.edu. Telephone: +12066853029.

the ever-growing demand for mathematical methods capable of handling *big data*. Two extremely successful mathematical strategies have been developed for handling data: the so-called *compressive sensing* (CS) and *machine learning* (ML) methods. In the former, sparsity in the data and/or representation of the data plays a defining role. In the latter, especially as it applies to pattern theory, clustering and dimensionality reduction, statistical learning,[11] and/or pattern classification/recognition[12] is accomplished by computing principal components (dimensionality reduction) of the dynamic data. Our objective is to integrate CS and ML together to enhance the modeling and predictive capabilities of complex dynamical systems. These two techniques (CS, ML) naturally partner together and are demonstrated to accurately characterize the flow around a cylinder. Although we will discuss ML and CS as separate methods, one can more broadly think of CS as simply another tool from the larger ML theoretical framework.

Compressed sensing allows the reconstruction of a signal using a small number of measurements, based on the fact that the signal has a sparse representation in an appropriate basis.[13–19] This allows the use of less measurements than the Nyquist–Shannon criterion requires.[20,21] Although the typical CS applications are found in areas spanning signal processing to computer vision, our objective is to use the methodology for the numerical solution of partial differential equations (PDE). We present a specific example – the incompressible Navier-Stokes flow around a cylinder. However, the framework presented can be applied to a much broader set of equations. At the method's core is fact that most complex systems, including the flow around a cylinder, exhibit an underlying low-dimensional structure in their dynamics. Unlike the standard application of CS for image processing, for example, where the sparse (wavelet) basis is already known, the optimal sparse basis considered here is computed directly from PDE simulations.

Historically, the problem of optimal basis selection is of paramount importance in complex dynamical systems. Recent works have already suggested some forms of statistical learning and ML. Indeed, sparsity with a pre-determined (non-optimal) basis can be used for PDE simulations to great advantage.[22–24] Moreover, similar statistical learning, library-based, and/or information theory methods[25,26] are used for basis selection in fluid dynamics, climate science,[27] and oceanography.[28] Such observations of dimensionality reduction allow compressive sensing to play a critical role in model building and assessment in the physical sciences.[29–33] Such a broad range of researchers and scientific topics demonstrates the growing importance of using and integrating the data-driven techniques of dimensionality reduction, machine/statistical learning, and compressive sensing. The current work combines all three to produce an effective strategy for characterizing complex systems.

To be more precise about the integration of CS and ML with fluid flows, consider a standard numerical scheme for solving PDEs by discretization (e.g., flow around a cylinder). That is, turn a PDE into a system of ordinary differential equations (ODE), then solve the latter equation with standard time-stepping tools, such as Runge-Kutta. For complex dynamical systems, the immediate difficulty is dimensionality – the dimension of the ODE depends on the number of points in the grid, which can be prohibitively large. This is due to the fact that an accurate PDE solution generally requires a large number of grid points, making the numerical solution computationally expensive. However, a well-known observation is that in many systems, the dynamics computed in the high-dimensional ODE, are observed to be low-dimensional. To this end, the proper orthogonal decomposition (POD) was developed where the dynamics is projected from the high dimensional discretized system to a low-dimensional system that can be solved with a significant computational improvement and often is amenable to analytic analysis. The specific projection (POD) is chosen based on the statistics obtained from solving the full system. Namely, a full simulation is first computed. Then a projection is computed that will preserve most of the data in the L_2-norm, i.e., the reduction and spatiotemporal encoding is accomplished using a singular value decomposition. Section III B elaborates on this method. For general information regarding the POD method and its application to the Navier-Stokes equation, see Berkooz, Holmes, and Lumley.[34]

Integration of the CS and ML schemes exploits the underlying low-dimensional (POD modes) dynamics observed in simulation and/or experiment. As stated, CS takes advantage of sparsity by considering a basis wherein the dynamics is sparse. Here, the sparse basis elements are the POD modes computed from full PDE simulations of the system. By characterizing various dynamical parameter regimes of the system, a library of sparse POD modes can be constructed, i.e., a supervised

ML algorithm is effectively enacted and the dynamics are *encoded*. Sparse measurements can then be used with the CS algorithm to identify, or *decode*, the correct POD modes for the given measurements, allowing for reconstruction of the dynamics. In our example of flow around a cylinder, the sparse POD basis elements are constructed from different Reynolds numbers where the dynamical behavior changes significantly. A limited number of pressure field measurements is all that is necessary to reconstruct the pressure field, and determine the current Reynolds number. To our knowledge, this proof of concept in integrating CS and ML with continuous complex systems is demonstrated here for the first time.

The structure of this paper is as follows: Sec. II offers an overview of the method advocated and highlights the key features of the data-driven modeling strategies, including how the dimensionality reduction techniques are used with sparse sensing. Section III describes the governing PDE system considered (the flow around a cylinder) and the numerical method used to study it, including its POD decomposition. Section IV specifies the implementation of the compressed sensing technique to the low-dimensional numerical solutions (ML library) of the Navier-Stokes equation. Section V contains the results of the CS and ML schemes for characterizing flow around the cylinder. Finally, Sec. VI contains a discussion of the method's future outlook.

II. OVERVIEW OF METHOD

Figure 1 helps illustrate the overall integration of the dimensionality reduction methods (machine learning) with sparse sensing in a complex system such as the flow around a cylinder. In particular,

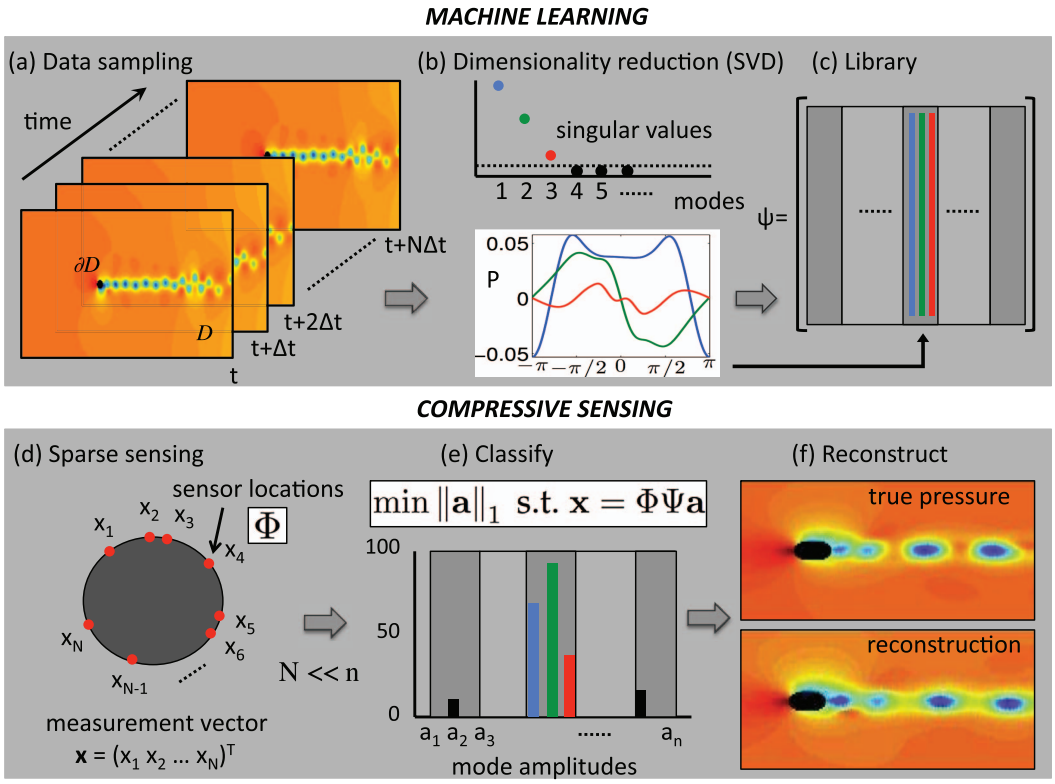

FIG. 1. Outline of algorithmic procedure used for dimensionally reducing, classifying, and reconstruction the flow around a cylinder. The procedure relies first on building a library of modes acquired, in an offline manner, by low-rank projections (SVD). The machine learning portion of the algorithm thus relies on (a) data sampling of the full, highly resolved complex system which is then (b) dimensionally reduced through SVD/POD/PCA and (c) stored in a library matrix Ψ. Once constructed, (d) sparse sensing at a limited number (N) of locations can be used to (e) classify the dynamical regime by solving a L_1 convex optimization problem. The L_1 promotes sparsity and allows the classification to identify the current relevant dynamic modes. Further, (e) the pressure field around the cylinder and in the flow domain can then be reconstructed.

the key steps involved in the machine learning portion, which is performed in an *offline* fashion, is separated from the compressive sensing implementation which is performed *online* given the ML results. Typically, the machine learning portion is extremely expensive from a computational viewpoint since it involves a refined and well-resolved characterization of the complex system across a broad range of parameter space. Thus, it is thought to be an offline computation. Once the highly resolved complex system has been dimensionally reduced, the compressive sensing scheme can take advantage of this as the representation of the dynamics are now easily represented in a sparse (low-dimensional) basis. This part of the computation can be done in an online, real-time manner. The following steps are critical for each portion of the method.

A. Machine learning (dimensionality reduction)

1. Data sampling and acquisition

The first step is to compute (or experimentally acquire) solutions of the complex system of interest as a function of a control/bifurcation parameter. For the flow around a cylinder, the goal is to compute the fluid pressure field as a function of time for various Reynolds numbers. Pressure data can be collected both on the cylinder itself ∂D and in the surrounding domain D. The respective data matrices $\mathbf{A}_{\partial D}$ and $\mathbf{A}_D$ are such that each column of the matrix is a snapshot of the pressure field at time t, $t + \Delta t$, $t + 2\Delta t$, $\cdots$, $t + M\Delta t$. Thus, there are $M + 1$ total snapshots (columns) in the data matrices. Note that 2D or higher-dimensional data are first reshaped into a column vector before adding it to the data matrix. The total number of snapshots and time between snapshots are chosen judiciously so as to properly characterize the fluid flow.[34] In the fluid flow example, the data can be taken simply on the boundary of the cylinder ∂D, or for future reconstructions of the flow field, in the domain surrounding the fluid D. Further discussion on this will be given at the end of Sec. V where reconstruction of the pressure field in D is achieved given sparse measurements on ∂D.

2. Dimensionality reduction and optimal (sparse) bases

The acquisition of data at various Reynolds numbers allows for the construction of the optimal (in a L^2 sense) bases modes associated with each Reynolds number. In particular, a POD reduction is accomplished with the singular value decomposition algorithm.[34] For each Reynolds number considered, the dominant pressure field modes on the cylinder and/or the surrounding fluid can be acquired. Typically, only those modes capturing a prescribed percentage of the energy (e.g., 99%) are kept. This gives a low-rank approximation of the underlying low-dimensional manifold on which the dynamics evolves.

3. Library

The collection of dominant POD modes are used to construct a library matrix Ψ. The goal is for the library to contain all possible or relevant low-rank approximation bases. Thus, this collection of optimal basis modes represents the most sparse representation of the dynamics for the various dynamical behaviors observed. It should be noted that the grouping of modes in Ψ that are generated at different Reynolds numbers are not orthogonal to each other. One could imagine reducing the library further by finding the correlation structures among the various dynamical modes using, for instance, the singular value decomposition (SVD), but our objective will be to use sparse sensing to classify and select the best modes associated with given sparse data measurements. This library building exercise is a very simplistic implementation of a ML scheme. But given its success, it suggests that more advanced ML techniques could potentially be applied with even greater success.

B. Compressive (sparse) sensing

1. Sparse measurement

A limited (sparse) number of pressure sensors are placed around the cylinder in order to classify and reconstruct the pressure field on the cylinder and in the flow field around it. The matrix Φ

prescribes the N measurement locations around the cylinder (as a projection). Each measurement x_j is arranged into a measurement vector $\mathbf{x}$. More precisely, given the full state vector $\mathbf{y}$ of pressure measurements around the cylinder, then $\mathbf{x} = \Phi\mathbf{y}$ specifies the relationship between the sparse measurements and the full state.

2. Classification

The primary role of the offline library building is now considered. Specifically, the first goal of the sparse measurement is to provide a basis for classifying the dynamical regime. Compressive sensing provides an ideal method for classification (or machine learning) given that the library building provided the sparse modes necessary for the compressive sensing framework. To be more precise, the CS scheme now considers solving the highly underdetermined system $\mathbf{x} = \Phi\Psi\mathbf{a}$ subject to finding the min $\|\mathbf{a}\|_1$. This results in a convex optimization problem where the role of the L^1 optimization is to promote sparsity in the vector $\mathbf{a}$ which determines the coefficient of each library mode element. In particular, almost all the vector components of $\mathbf{a}$ will be identically zero. The remaining non-zero modes are those associated with the dynamical regime currently measured by the sparse sensors.

3. State estimation and reconstruction

Once the sensors have classified the current dynamical regime of the system, the pressure field both around the cylinder ∂D and in the fluid domain D can be reconstructed from the library data. Note that once the appropriate modes are classified, a L_2 projection (pseudo-inverse) is used to project the measurements onto these selected modes and the pressure field on the cylinder and around the cylinder can be constructed. Thus, the sparse sensing is used for classification only. There are several options how to do this procedure, we discuss one of them at the end of Sec. V. One could also then use this reconstruction as the initial state of a Galerkin POD projection aimed at predicting the future state of the system.[34] Thus, there are three potential tasks that can be achieved with such an integration of ML and CS: classification, reconstruction, and a low-rank projection of the future state.

The above outlines a general procedure that can be broadly used in complex systems. What follows is a specific demonstration of this technique to the flow around a cylinder. Its success emphasizes the role that data-driven methods such as ML and CS can have in quantifying dynamical systems. Many of the more advanced classification, clustering, dimensionality reduction, and sparse sensing schemes are currently under consideration.

III. GOVERNING EQUATION AND NUMERICAL METHODS

To demonstrate the integration of ML and CS on complex dynamical systems, we study the flow of an incompressible fluid around a cylinder in a channel. This system is described by the Navier-Stokes equation

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla\mathbf{p} - \nu\nabla^2\mathbf{u} = \mathbf{0}, \tag{1a}$$

$$\nabla\cdot\mathbf{u} = \mathbf{0}, \tag{1b}$$

where $\mathbf{u}(\mathbf{x}, \mathbf{t}) \in \mathbf{R^2}$ represents the two-dimensional (2D) velocity of the fluid, and $p(x, t) \in R$ is the corresponding pressure. The boundary conditions are as follows:

1. Constant flow of $\mathbf{u} = (\mathbf{1}, \mathbf{0})^{\mathbf{T}}$ at $x = -15$, i.e., the entry of the channel.
2. Constant pressure of $p = 0$ at $x = 25$, i.e., the end of the channel.
3. Neumann boundary conditions, $\frac{\partial \mathbf{u}}{\partial \mathbf{n}} = 0$ on the boundary of the channel.
4. No-slip conditions, $\mathbf{u} = \mathbf{0}$ on the boundary of the cylinder.

This system is chosen because it is a well-known and classic problem of applied mathematics. Furthermore, it has been demonstrated to exhibit low-dimensional behavior as the parameter ν, the Reynolds number, is varied.

The characteristic dynamics of the fluid flow as a function of the Reynolds number is beautifully visualized, for instance, in Van Dykes *An Album of Fluid Motion.*[35] At the lowest Reynolds numbers, the fluid is described by a simple laminar flow. Increasing the Reynolds number produces the first non-trivial dynamics which is characterized by the creation of two circulating regions along with a stagnation point behind the cylinder. Further increasing the Reynolds number induces the onset of periodic, von Karman vortex shedding. Finally, further increasing the flow rate beyond a critical level produces fully developed turbulence in the wake of the cylinder. As might be expected, the first three dynamical regimes produce low-dimensional representations of the fluid motion. But remarkably, even in the fully turbulent regime, snapshot based methods for $O(1)$ samples of time show the fluid flow to be relatively low-dimensional.[34]

A. Numerical methods

The numerical solution of the incompressible Navier-Stokes equation is computed using the Nektar++ package based on Ref. 36, which produces accurate results and uses a high-order spectral element method with element order 12. The mesh used is a non-uniform mesh consisting of 412 elements, and the high-order simulation produces high-resolution data at the 66 000 Gaussian quadrature points. Figure 2 shows the grid used for computation of the fluid flow. Note the mesh refinement around and to the right of the cylinder where the most interesting fluid flow occurs. In contrast, no refinement is required far to the left, top, and bottom of the computational domain. The mesh used is sufficient for demonstrating the key concepts of ML and CS integration. In particular, the Reynolds number range considered is confined to between 40 and 3000. Higher values of Reynolds numbers may require further refinement of the mesh.

Simulations were generated for the Reynolds numbers $\nu = 40$, 150, 300, 1000, and 3000 with arbitrary initial conditions. The dynamics were simulated until a persistent behavior, stationary or periodic, was observed for the non-turbulent dynamics. Upon achieving a steady- or periodic-state, the dynamics was sampled through a series of snapshots whose time between intervals was much faster than the periodicity observed in the dynamics. Figures 3 and 4 show a typical snapshot of the streamlines and the pressure profiles taken once the system was in the equilibrium or time-periodic state. Note that the method is not geared to capture transient dynamics associated with rapid changes in the Reynolds number.

To attempt a pressure field reconstruction, we consider only data from the perimeter of the cylinder. In particular, one can place pressure sensors on the cylinder to reconstruct the flow and pressure fields. However, placing sensors in the fluid's flow itself may be valid for lab experiments and computer simulations alone as no realistic system of interest, such as an airfoil or insect wing, places sensors directly in the flow field behind the structure.

Figure 5 shows the time dynamics of the pressure field around the cylinder for several Reynolds numbers. As can be seen, increasing the Reynolds number drives the dynamics from a steady-state

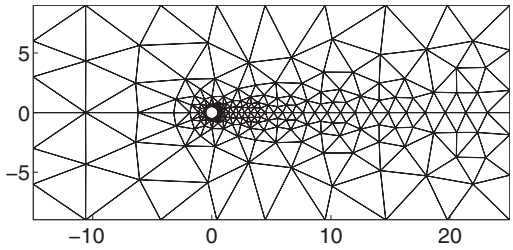

FIG. 2. The grid used for simulating the Navier-Stokes equation is a non-structured grid containing 412 elements. Mesh refinement is applied around the cylinder and in its wake, where complex fluid flow is observed. Using a high order solver around 66 000 samples are computed in each snapshot.

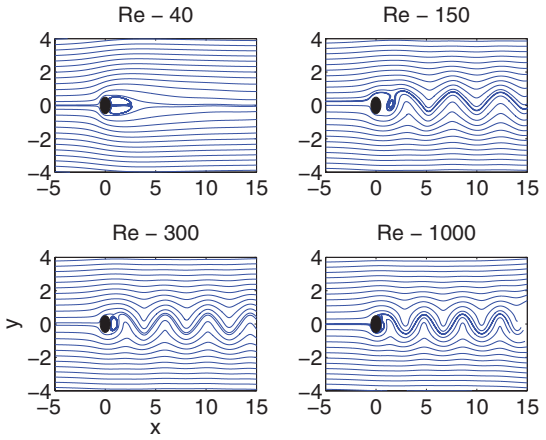

FIG. 3. Stream lines near the cylinder for a typical snapshot taken for Reynolds numbers 40, 150, 300, and 1000 (clockwise).

configuration to a time-periodic evolution to a chaotic (turbulent) evolution. The waterfall plots of the pressure dynamics on the cylinder are essentially the snapshots used for projecting the dynamics onto a low-dimensional manifold, where the dynamics occurs as described by the POD methodology.[34]

B. Proper orthogonal decomposition

The POD method is used to approximate the dynamics of a PDE with a low-dimensional ODE. Moreover, it attempts to construct (and is guaranteed to do so) the *best* dimensionality reduction (low-rank approximation) possible in a L_2 sense. The reduction is accomplished by employing the actual computational (or potentially experimentally measured) solution and taking the linear projection that maximizes the amount of energy preserved, i.e., the original data's L_2-norm. Essentially, the singular value decomposition can be applied to snapshots of the simulation data. This method can be seen as a Galerkin method, where instead of using the arbitrary Fourier mode basis, the basis is chosen with respect to the specific problem based on the statistical properties of the solution. The POD reduction, perhaps in a slightly modified form, is known throughout the literature alternatively as principal component analysis (PCA), empirical orthogonal functions (EOF), the Hotelling transform, or a Karhunen-Loéve expansion. Here, we will simply refer to it as POD.

The application of the POD method to the study of channel flow around a cylinder was first presented by Deane, Kevrekidis, Karniadakis, and Orszag.[37] In that study, the solution of the

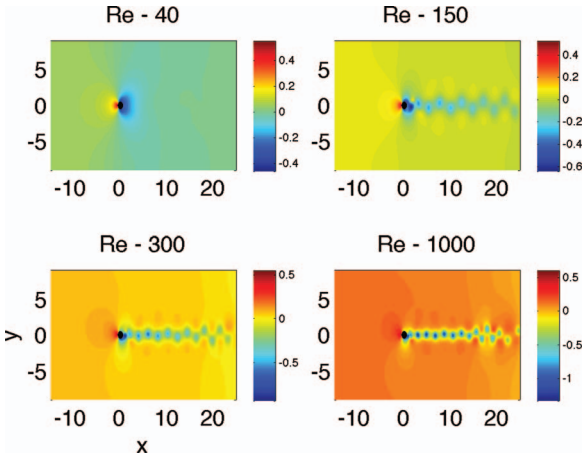

FIG. 4. Pressure field snapshot for Reynolds numbers 40, 150, 300, and 1000.

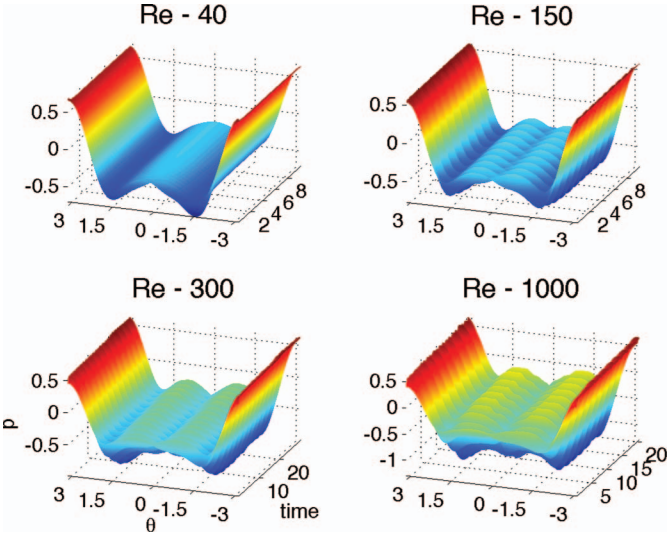

FIG. 5. Pressure field around the cylinder for increasing Reynolds number 40, 150, 300, and 1000. As the Reynolds number increases and dynamical evolution becomes more complicated, it requires a larger number of POD modes to represent the dynamics. Regardless, the pressure field remains low dimensional for all cases, requiring 1, 4, 6, and 8 modes to capture 99.9% of the energy, respectively.

low-dimensional system is shown to approximate quite well the original system in the non-turbulent regime. The POD method was also applied to the study of three-dimensional cylinder flow by Ma and Karniadakis[38] and with control systems by Bergmann, Cordier, and Brancher.[39] In this work, we do not use POD modes to compute the system's dynamics, but rather as the basis that represents our data in a sparse manner, i.e., the POD modes are the library elements that encode the low-rank dynamics of the system for a range of Reynolds numbers.

To compute the POD modes for the Reynolds numbers we are interested in, namely, $\nu = 40$, 150, 300, 800, 1000, and 3000, we simulate the dynamics, with a fixed Reynolds number, until we reach a steady or periodic state. Data sampling is then achieved by taking snapshots of the simulation results. Recall that only the pressure field on the cylinder is measured. We perform a singular value decomposition on the data to obtain a low-dimensional representation, keeping only those POD modes that capture a prescribed percentage of the energy. As noted, the flow characteristics are low rank. Thus, the dynamics can be thought of as sparse in the POD basis. Indeed, validation of our notion of data sparseness is demonstrated by the decay of singular values in Fig. 6. This figure demonstrates the primary thrust of the article: (i) the dynamics throughout a large range of Reynolds numbers always can be represented by a small number of appropriately chosen modes, (ii) the dynamics is sparse in the POD basis representation, and (iii) a library constructed from the POD modes at different Reynolds numbers captures the full range of dynamics possible for the flow around a cylinder (aside from the fully turbulent regime).

For the POD mode selection, the dimension of the projected, low-rank system must be specified. This is chosen according to the data by selecting a threshold of 99.9% of the energy we want to conserve with the projection. The cutoff threshold is rather arbitrary and a more rigorous analysis characterizing the role of the threshold on POD performance and convergence is still lacking. Regardless, with a given threshold our objective is to select the minimal number of POD modes needed for each Reynolds number. Once all of the POD modes are computed in the preprocessing stage, i.e., for the various Reynolds numbers, we place them as the column vectors in the matrix Ψ. Figure 7 shows the three most dominant modes for various Reynolds numbers. The matrix Ψ is our library. The library encapsulates all known behaviors of the system, or, at least, those explored via computational simulations. Success of the library-based method is based upon appropriately sampling all relevant dynamics and parameter ranges under which the system will be considered.

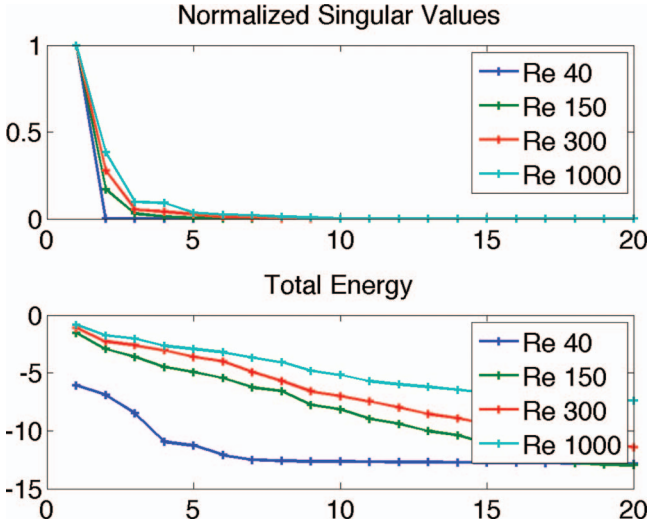

FIG. 6. The top plot presents the decay of singular values of the data matrix, for various Reynolds number. The graph presents the singular values normalized by the largest singular value. The bottom graph presents, in logarithmic scale, how much energy is preserved by projecting the dynamics to a n-dimensional space, for $n = 1, 2, \ldots, 20$. As the Reynolds number increases, a higher dimensional space is needed to preserve the same amount of energy.

To summarize the preprocessing stage and outline the underlying algorithm developed, we performed the following steps for the Reynolds numbers $\nu = 40$, 150, 300, 800, 1000, and 3000:

1. Simulate the dynamics with Nektar++ until a steady- or periodic-state is reached.
2. Once initial transients die out and the solution of interest is reached, sample the pressure on the cylinder's perimeter through a series of snapshots. This corresponds to our sensors.
3. Generate the data matrix $\mathbf{A}^{\nu}_{\partial D}$ containing the time snapshots as its columns. Note that the superscript ν denotes the data matrix for a given Reynolds number. Note that data matrix $\mathbf{A}^{\nu}_{D}$ for the pressure field surrounding the cylinder is also collected for future reconstruction purposes. A dimensionality reduction and library building procedure for $\mathbf{A}^{\nu}_{D}$ is not performed.
4. Perform a singular value decomposition on the data matrix, $\mathbf{A}^{\nu}_{\partial D} = \mathbf{U}^{\nu}\mathbf{S}^{\nu}\,(\mathbf{V}^{\nu})^{*}$, where $\mathbf{V}^{\nu}$ and $\mathbf{U}^{\nu}$ are unitary, $\mathbf{S}^{\nu}$ is diagonal containing non-negative entries, and $(\mathbf{V}^{\nu})^{*}$ denotes the matrix conjugate to $\mathbf{V}^{\nu}$.
5. Identify the d^{ν} most energetic POD modes so their total energy will comprise 99.9% of the data matrix, and dimensionally reduce the system appropriately to this dimension. A matrix $\bar{\mathbf{V}}^{\nu}$ is created containing the relevant POD modes as its columns.

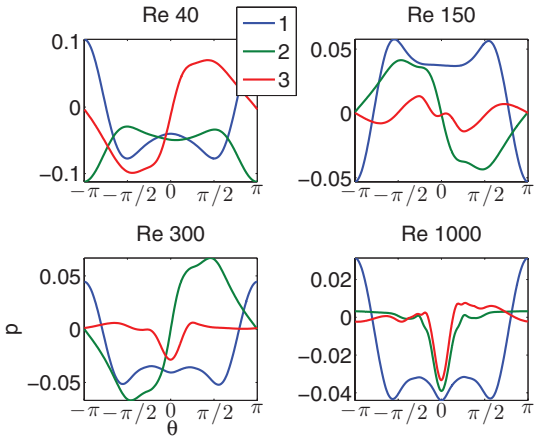

FIG. 7. The three top eigenfunctions, namely, the three most dominant modes, are presented for various Reynolds numbers.

Once this process is completed at all the desired Reynolds numbers, we concatenate the $\bar{\mathbf{V}}^{\nu}$ matrices to generate our library matrix Ψ, namely,

$$\Psi = \left[\bar{\mathbf{V}}^{40}, \bar{\mathbf{V}}^{150}, \bar{\mathbf{V}}^{300}, \bar{\mathbf{V}}^{800}, \bar{\mathbf{V}}^{1000}, \bar{\mathbf{V}}^{3000}\right]. \tag{2}$$

Of note, truncation to a lower-dimensional system described in step 5 is accomplished by picking the most energetic mode to constitute, for example, 99.9% of the energy. Particularly, if we denote the diagonal of $\mathbf{S}$ by $s_1 \geq s_2 \geq \cdots \geq s_n$, then we pick the smallest d satisfying

$$r = \frac{s_1^2 + s_2^2 + \cdots + s_d^2}{s_1^2 + s_2^2 + \cdots + s_n^2} \geq 99.9\%. \tag{3}$$

This affords a simple method for determining the rank of the projection space to be considered, thus determining the number of POD modes to keep in $\bar{\mathbf{V}}^{\nu}$. With the library's completion, the addition of a compressive sensing scheme for accessing the library is now required. Note that the implementation of the preceding algorithm is equivalent to a supervised ML algorithm, where low-dimensional structures in the computational data are discovered and saved for future statistical testing purposes.

IV. COMPRESSED SENSING WITH MACHINE LEARNED LIBRARY ELEMENTS

The compressive scheme applied here deals with solving an underdetermined linear systems of equations. Overdetermined systems are typically solved by minimizing the error between the fitted and the actual data, mostly using the pseudo-inverse operator, that minimizes the L_2-norm of this error. In contrast, the entire objective of compressed sensing is to obtain a sparse (sparsest) solution to the underdetermined system, from the set of all possible solutions. In theory, the minimization of the L_0-norm of the system's solution is ideal for promoting sparseness. However, minimizing the L_0-norm is NP-hard and thus, computationally intractable. As an alternative, it has been shown that using the L_1 norm instead of L_0 results in both (i) a computationally tractable problem via convex optimization and (ii) sparse solutions. Indeed, the L_1 norm can simply be thought of as a proxy for sparsity. Baraniuk[19] illustrate these ideas beautifully in a set of lecture notes in IEEE Signal Processing Magazine (Lecture Notes), and the reader is encouraged to review this paper if unfamiliar with sparse approximation and compressive sensing.

This problem is stated more formally as follows: Minimize the zero-norm of $\mathbf{a}$

$$\|\mathbf{a}\|_0 = card\left(\{i | a_i \neq 0\}\right) \tag{4}$$

restricted to the linear constraint

$$\mathbf{x} = \Phi\Psi\mathbf{a}, \tag{5}$$

where Φ specifies the sparse measurements to be taken.[19] As already stated, solving this problem as it appears is computationally intractable because it is NP-hard. Fortunately, Candes, Romberg, and Tao[14] show that under general conditions, the resulting parameters can be obtained by minimizing the L_1 norm of $\mathbf{a}$ rather than the L_0 norm. This turns the problem into the computationally tractable convex optimization problem

$$\min \|\mathbf{a}\|_1 \ \text{ s.t. } \mathbf{x} = \Phi\Psi\mathbf{a}. \tag{6}$$

The main assumption when applying compressed sensing is that the given vector $\mathbf{x} = \Phi\Psi\mathbf{a}$ was generated from a sparse vector $\mathbf{a}$. Thus, we need to find a basis that represents our data in a sparse manner. The basis described in this paper is obtained by the POD method, as it is optimal in approximating the data in the L_2 sense using a small number of modes. Namely, our assumption is that given a snapshot of the dynamics obtained from an unknown Reynolds number, and all of the dominant POD modes, precomputed for several different Reynolds numbers, the sparsest representation of the data is obtained using vectors corresponding to an actual unknown parameter. To this end, our previously built library of vectors (POD modes) in our matrix Ψ is the sparse basis required for the previously described CS scheme.

To solve the L_1 minimization problem, a convex optimization routine is required. Here, we implement the solution of (6) using CVX, a package that can be implemented with MATLAB for

specifying and solving convex programs[40] (CVX: Matlab software for disciplined convex programming, version 2.0 beta, September 2012). Once the POD library of modes (Ψ) is constructed, new pressure measurements can be used, and the minimization problem (6) can be solved using a simple laptop running MATLAB.

V. RESULTS

As a first proof-of-concept demonstration of the integration of CS and ML to characterize fluid flow, the methods are used to compute the Reynolds number of the flow field from a sparse number of measurements as the flow field changes in time. This Reynolds number reconstruction was applied to a numerically generated flow, where the Reynolds number varies slowly after reaching a steady state. Specifically, we generated a flow that starts with Reynolds number 40 and jumps successively to 150, 300, and 1000. The change in Reynolds number occurs after a steady-state/periodic-state is achieved. The CS scheme is used to reconstruct the Reynolds number at every snapshot in time where data are procured.

Recall that data reconstruction from a single snapshot is performed by reducing the dimension of our data through sparse measurements specified by the matrix Φ,[19] i.e., pressure measurements on the perimeter of the cylinder. We present two different methods for this dimension reduction:

1. Applying a random projection by a rectangular Gaussian matrix Φ of dimension N.[13–16]
2. Picking a small subset of the sensors and setting Φ accordingly. This is done with two different strategies:
   (a) Randomly choosing a small subset of N sensors at random.
   (b) Selecting N sensors according to the maxima and minima position of the POD modes with the highest energies in our library. The latter method was first suggested by Cohen, Siegel, and McLaughlin[41] for a different goal, namely, the reconstruction of the flow based on a small number of samples. This was further developed by Yildirim, Chryssostomidis, and Karniadakis[42] for ocean modeling.

For each method, we present our results for $N = 10$ and $N = 20$, the dimension to which we reduce the problem. The ratio of energy we choose to obtain our POD modes is set to 99.9% as it provides good results.

The classification of the Reynolds number is done by solving the optimization problem in (6) and obtaining the coefficient vector **a**. Each entry in **a** corresponds to the energy of a single POD mode from our library. Because the optimization in L_1 promotes sparsity, only those coefficients from POD modes associated with the measured flow remain nonzero. The classification of the Reynolds number is done by summing the absolute value of the coefficient that corresponds to each Reynolds number. To account for the large number of coefficients allocated for the higher Reynolds number (which may be 16 POD modes, rather than a single coefficient for Reynolds number 40), we divide by the square root of the number of POD modes allocated in **a** for each Reynolds number. The classified method is the one that has the largest magnitude after this process. The result of this classification process is summarized in Table I. Seeing that selecting the data-driven sensor locations (in (b)) obtains the best results, we decided to further investigate the performance as a function of the number of sensors. This can be seen in Figure 8. Although the accuracy in classification is quite

TABLE I. Success rate of classification using N sensors with library modes selected to have $r = 99.9\%$ or 99.99% as given in (3). For a small number of sensors ($N = 10$) and $r = 99.9\%$, all classification schemes fall below 80%. However, as the number of sensors and energy cutoff increase, the POD based sensor selection provides the best classification possibilities.

	Random projection (%)	N-random sensors (%)	N-sensors by POD (%)
$N = 10, r = 99.9\%$	79.5	78.9	75.4
$N = 10, r = 99.99\%$	65.0	48.7	76.9
$N = 20, r = 99.9\%$	78.1	76.1	92.9
$N = 20, r = 99.99\%$	82.3	85.2	88.3

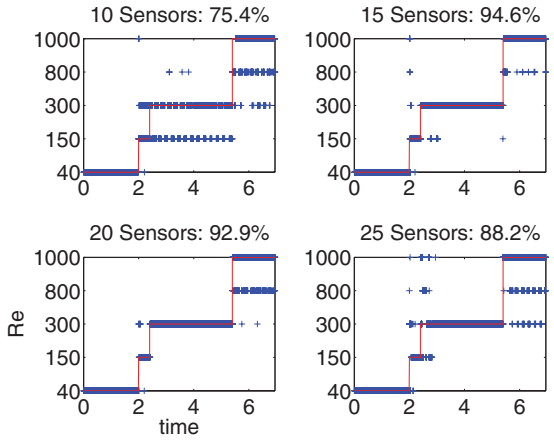

FIG. 8. Reynolds number classification for a time-varying flow, using a different number of input sensors. In each panel, you can see the success rate given the number of sensors, and also the actual Reynolds number used in the full simulation (solid line) along with its compressive sensing identification (crosses).

high, many of the false classifications are due to categorizing a Reynolds number flow of 1000 as a Reynolds number flow of 800. This is largely due to the fact that these two Reynolds numbers are strikingly similar so that the algorithm proposed has a difficult time separating their modal structures.

Finally, to visualize the entire sparse sensing and reconstruction process more carefully, Fig. 9 shows both the Reynolds number reconstruction for the time-changing flow field along with the pressure field and flow field reconstructions at select locations in time. Note that the CS scheme along with the supervised ML library provide an effective method for characterizing the flow strictly through sparse measurements. For higher Reynolds numbers, it becomes much more

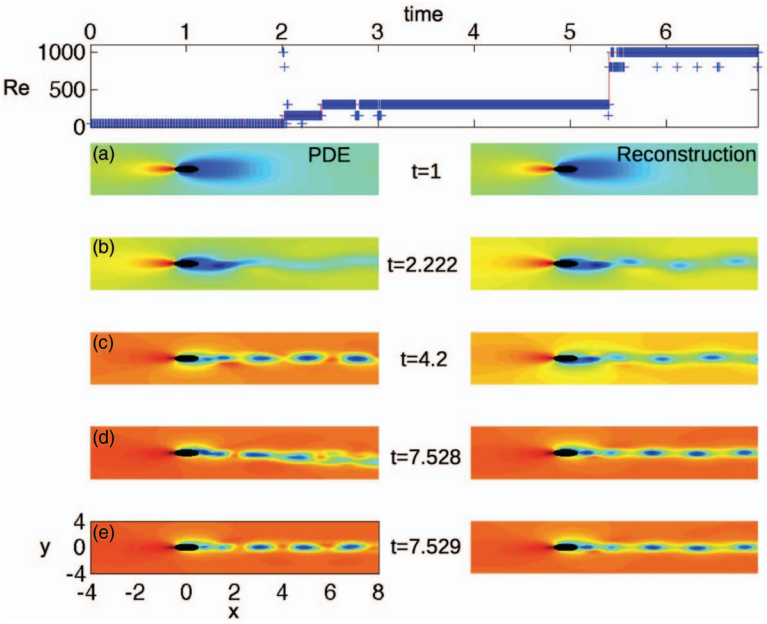

FIG. 9. Sparse-sensing Reynolds number identification and pressure-field reconstruction for a time-varying flow. The top panel shows the actual Reynolds number used in the full simulation (solid line) along with its compressive sensing identification (crosses). Panels (a)–(d) show the reconstruction of the pressure field at four different locations in time (top panel) demonstrating an accurate (qualitatively) reconstruction of the pressure field. (The left side the simulated pressure field is presented, while the right side contains the reconstruction.) Note that for higher Reynolds numbers, the classification becomes more difficult.

difficult to accurately classify the flow field with such a small number of sensors. However, this does not necessarily jeopardize the ability to reconstruct the pressure field as many of the library elements at higher Reynolds numbers are fairly similar.

It should be noted that reconstruction of the pressure field on the cylinder can easily be achieved once the classification step is done. In particular, one simply projects the data measurements onto the dominant modes of ∂D for the given Reynolds number found from classification. Thus, this is a fairly simple exercise in reconstruction by using an L_2 measure and a pseudo-inverse projection. Reconstruction of the pressure field in the domain D surrounding the cylinder can also be done this way. However, the alternative method we outline here has a tremendous advantage: it does not require performing a SVD of the full pressure field data matrix $\mathbf{A}_D^\nu$ in the domain D (see Fig. 1). It is highly desirable and advantageous to avoid performing the SVD of such large data sets. Consider instead the decomposition of the collected data on the boundary and domain: $\mathbf{A}_{\partial D}^\nu = \mathbf{U}^\nu \mathbf{S}^\nu (\mathbf{V}^\nu)^*$ and $\mathbf{A}_D^\nu = \mathbf{U}^\nu \mathbf{S}^\nu \left(\tilde{\mathbf{V}}^\nu\right)^*$, respectively. Note that these decompositions share common $\mathbf{U}^\nu$ (time evolution) and $\mathbf{S}^\nu$ (singular values) matrices. Indeed, the difference between these two decompositions simply is related to their modal structures which lie on the cylinder, $(\mathbf{V}^\nu)^*$, or in the surrounding fluid, $\left(\tilde{\mathbf{V}}^\nu\right)^*$. Given this fact, the modes in the surrounding fluid can be found from

$$\left(\tilde{\mathbf{V}}^\nu\right)^* = (\mathbf{S}^\nu)^{-1} (\mathbf{U}^\nu)^* \mathbf{A}_D. \tag{7}$$

Thus, the reconstruction of the pressure field modes around the cylinder reduces to a matrix multiplication on the data collected $\mathbf{A}_D$. Reconstruction can then be achieved by projecting the sparse measurements onto these modes. This avoids the costly computational operation associated with the SVD procedure on the data $\mathbf{A}_D$.

VI. CONCLUSIONS AND DISCUSSION

Modern data analysis techniques are providing transformative algorithms and methods for exploring complex systems such as fluid flows. In the context of exploring high-dimensional dynamical systems, a variety of techniques have successfully been applied towards developing low-dimensional models capable of greatly improving the analysis, control, and predictability of a given model. The two promising data analysis techniques of ML and CS have hitherto been unexplored in the context of dynamical systems theory. When used in conjunction, ML and CS provide an ideal framework for using sparse sensors in a complex dynamical system for reconstruction of the dynamics and evolution of the system on an optimal low-dimensional manifold. In the fluid flow application advocated here, the underpinnings of bio-inspired flight control and stabilization are suggested. This is of particular importance in understanding the robust flight stabilization mechanism used for improving propulsion efficiency, maximizing thrust and lift, and increasing maneuverability in, for example, insects, fish, and bird.

The combination of dimensionality reduction, CS and ML advocated here can be applied in an exceptionally broad context. Indeed, such algorithmic strategies can be used to enhance computation and efficiently identify measurement locations in a given system by *remembering* the system's key characteristics. The low-rank nature of the approximations used allows construction of control algorithms that wrap around the dimensionality reduction, CS, and ML infrastructure—another appealing aspect of the methodology. For example, in flight control, the identification of the dominant dynamical modes through a sparse set of measurements would only be the first step in applying a control algorithm for stabilizing a constant lift. As such, the proposed method would serve as the internal portion of the control algorithm. This will be explored further in future work. Further considerations will also be given to sampling in both time and space with the goal of reducing the number of sensors required for flow-field identification and reconstruction even further.

ACKNOWLEDGMENTS

We are indebted to Daniele Venturi for invaluable support in implementing the Nektar++ software package. J. N. Kutz acknowledges valuable conversations relating to this work with Jonathan

Tu and Steven Brunton. The work of I. Bright and G. Lin was supported by the Applied Mathematics Program within the (U.S.) Department of Energy (DOE) Office of Advanced Scientific Computing Research (ASCR). Pacific Northwest National Laboratory is operated by Battelle Memorial Institute for the (U.S.) Department of Energy under Contract No. DE-AC05-76RL01830. J. N. Kutz acknowledges support from the National Science Foundation (NSF) (DMS-1007621) and the (U.S.) Air Force Office of Scientific Research (USAFOSR) (FA9550-09-0174).

[1] J. J. Allen and A. J. Smits, "Energy harvesting eel," J. Fluids Struct. **15**, 629 (2001).
[2] R. Clark and A. J. Smits, "Thrust production and wake structure of a batoid-inspired oscillating fin," J. Fluid Mech. **562**, 415 (2006).
[3] J. C. Nawroth, H. Lee, A. W. Feinberg, C. M. Ripplinger, M. L. McCain, A. Grosberg, J. O. Dabiri, and K. K. Parker, "A tissue-engineered jellyfish with biomimetic propulsion," Nat. Biotechnol. **30**, 792 (2012).
[4] R. W. Whittlesey, S. C. Liska, and J. O. Dabiri, "Fish schooling as a basis for vertical-axis wind turbine farm design," Bioinspiration Biomimetics **5**, 035005 (2010).
[5] J. Birch and M. Dickinson, "Spanwise flow and the attachment of the leading-edge vortex on insect wings," Nature (London) **412**, 729 (2001).
[6] J. O. Dabiri, "Optimal vortex formation as a unifying principle in biological propulsion," Annu. Rev. Fluid Mech. **41**, 17 (2009).
[7] S. P. Sane, "The aerodynamics of insect flight," J. Exp. Biol. **206**, 4191 (2003).
[8] M. J. Shelley and Z. J. Zhang, "Flapping and bending bodies interacting with fluid flows," Annu. Rev. Fluid Mech. **43**, 449 (2011).
[9] Z. J. Wang, "Dissecting insect flight," Annu. Rev. Fluid Mech. **37**, 183 (2005).
[10] T. Y. Wu, "Fish swimming and bird/insect flight," Annu. Rev. Fluid Mech. **43**, 25 (2011).
[11] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer, 2009).
[12] R. Duda, P. Hart, and D. Stork, *Pattern Theory* (Wiley, 2001).
[13] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," IEEE Trans. Inf. Theory **52**(2), 489–509 (2006).
[14] E. J. Candes, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," Commun. Pure Appl. Math. **59**(8), 1207–1223 (2006).
[15] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" IEEE Trans. Inf. Theory **52**(12), 5406–5425 (2006).
[16] D. L. Donoho, "Compressed sensing," IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006).
[17] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," IEEE Trans. Inf. Theory **53**, 4655–4666 (2007).
[18] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," IEEE Trans. Inf. Theory **56**(4), 1982–2001 (2010).
[19] R. Baraniuk, "Compressive sensing," *IEEE Signal Processing Magazine (Lecture Notes)* **24**(4), 118–124 (2007).
[20] H. Nyquist, "Certain topics in telegraph transmission theory," Trans. Am. Inst. Electr. Eng. **47**(2), 617–644 (1928).
[21] C. E. Shannon, "A mathematical theory of communication," Bell Syst. Tech. J. **27**, 379–423 (1948).
[22] M. Farge, K. Schneider, and N. Kevlahan, "Non-Gaussianity and coherent vortex simulation for two-dimensional turbulence using an adaptive orthogonal wavelet basis," Phys. Fluids **11**, 2187 (1999).
[23] N. Okamoto, K. Yoshimatsu, K. Schneider, M. Farge, and Y. Kaneda, "Coherent vortices in high resolution direct numerical simulation of homogeneous isotropic turbulence: A wavelet viewpoint," Phys. Fluids **19**, 115109 (2007).
[24] H. Schaeffer, S. Osher, R. Caflisch, and C. Hauck, "Sparse dynamics of partial differential equations," Proc. Natl. Acad. Sci. U.S.A. **110**, 6634–6639 (2013).
[25] R. R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection," IEEE Trans. Inf. Theory **38**(2), 713–718 (1992).
[26] T. Wager, L. Atlas, L. Leotti, and J. Rilling, "Predicting individual differences in placebo analgesia: Contributions of brain activity during anticipation and pain experience," J. Neurosci. **31**, 439–452 (2011).
[27] D. Giannakis, A. Majda, and I. Horenko, "Information theory, model error, and predictive skill of stochastic models for complex nonlinear systems," Physica D **241**, 1735 (2012).
[28] A. Agarwal and P. F. J. Lermusiaux, "Statistical field estimation for complex coastal regions and archipelagos," Ocean Model. **40**, 164–189 (2011).
[29] L. J. Nelson, G. L. Hart, F. Zhou, and V. Ozoliņš, "Compressive sensing as a paradigm for building physical models," Phys. Rev. B **87**, 035125 (2013).
[30] A. Shabani, R. L. Kosut, M. Mohseni, H. Rabitz, M. A. Broome, M. P. Almeida, A. Fedrizzi, and A. G. White, "Efficient measurement of quantum dynamics via compressive sensing," Phys. Rev. Lett. **106**, 100401 (2011).
[31] W. X. Wang, R. Yang, Y. C. Lai, V. Kovanis, and C. Grebogi, "Predicting catastrophes in nonlinear dynamical systems by compressive sensing," Phys. Rev. Lett. **106**, 154101 (2011).
[32] A. B. Tayler, D. J. Holland, A. J. Sederman, and L. F. Gladden, "Exploring the origins of turbulence in multiphase flow using compressed sensing MRI," Phys. Rev. Lett. **108**, 264505 (2012).
[33] E. Candes, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" J. ACM **58**(11), 1–37 (2011).
[34] G. Berkooz, P. Holmes, and J. L. Lumley, "The proper orthogonal decomposition in the analysis of turbulent flows," Annu. Rev. Fluid Mech. **25**, 539–575 (1993).
[35] M. Van Dyke, *An Album of Fluid Motion* (Parabolic Press, 1982).

[36] G. E. Karniadakis, and S. J. Sherwin, "Spectral/hp element methods for computational fluid dynamics," *Numerical Mathematics and Scientific Computation* (Oxford University Press, New York, 2005).
[37] A. E. Deane, I. G. Kevrekidis, G. E. Karniadakis, and S. A. Orszag, "Low- dimensional models for complex geometry flows: Application to grooved channels and circular cylinders," Phys. Fluids A **3**(10), 2337–2354 (1991).
[38] X. Ma and G. E. Karniadakis, "A low-dimensional model for simulating three-dimensional cylinder flow," J. Fluid Mech. **458**, 181–190 (2002).
[39] M. Bergmann, L. Cordier, and J. P. Brancher, "Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model," Phys. Fluids **17**(9), 097101 (2005).
[40] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences Vol. 371, edited by V. Blondel, S. Boyd, and H. Kimura (Springer-Verlag Limited, 2008), pp. 95–110.
[41] K. Cohen, S. Siegel, and T. McLaughlin, "Sensor placement based on proper orthogonal decomposition modeling of a cylinder wake," in *Proceedings of the 33rd AIAA Fluid Dynamics Conference and Exhibit, Orlando, FL, 23–26 June 2003*.
[42] B. Yildirim, C. Chryssostomidis, and G. E. Karniadakis, "Efficient sensor placement for ocean measurements using low-dimensional concepts," Ocean Model. **27**(3), 160–173 (2009).