Sparse Reconstruction of Systems of Ordinary Differential Equations

Manuel Mai\textsuperscript{a}, Mark D. Shattuck\textsuperscript{b,c}, Corey S. O’Hern\textsuperscript{c,a,d,e,}\textsuperscript{*}

\textsuperscript{a}Department of Physics, Yale University, New Haven, Connecticut 06520, USA
\textsuperscript{b} Benjamin Levich Institute and Physics Department, The City College of New York, New York, New York 10031, USA
\textsuperscript{c} Department of Mechanical Engineering & Materials Science, Yale University, New Haven, Connecticut 06520, USA
\textsuperscript{d} Department of Applied Physics, Yale University, New Haven, Connecticut 06520, USA
\textsuperscript{e} Graduate Program in Computational Biology and Bioinformatics, Yale University, New Haven, Connecticut 06520, USA

Abstract

We develop a numerical method to reconstruct systems of ordinary differential equations (ODEs) from time series data without a priori knowledge of the underlying ODEs and without pre-selecting a set of basis functions using sparse basis learning and sparse function reconstruction. We show that employing sparse representations provides more accurate ODE reconstruction compared to least-squares reconstruction techniques for a given amount of time series data. We test and validate the ODE reconstruction method on known 1D, 2D, and 3D systems of ODEs. The 1D system possesses two stable fixed points; the 2D system possesses an oscillatory fixed point with closed orbits; and the 3D system displays chaotic dynamics on a strange attractor. We determine the amount of data required to achieve an error in the reconstructed functions to less than 0.1%. For the reconstructed 1D and 2D systems, we are able to match the trajectories from the original ODEs even at long times. For the 3D system with chaotic dynamics, as expected, the trajectories from the original and reconstructed systems do not match at long times, but the reconstructed and original models possess similar Lyapunov exponents. Now that we have validated this ODE reconstruction method on known models, it can be employed in future studies to identify new systems of ODEs using time series data from deterministic systems for which there is no currently known ODE model.

Keywords: Viral diseases, Immune system diseases, Data analysis: algorithms and implementation, Time series analysis, Nonlinear dynamics and chaos

PACS: 87.19.xd, 87.19.xw, 07.05.Kf, 05.45.Tp, 05.45.-a

1. Introduction

We present a methodology to generate systems of ordinary differential equations (ODEs) that can accurately reproduce measured time series data. In the past, physicists have constructed ODEs by writing down the simplest mathematical expressions that are consistent with the symmetries and fixed points of the physical system. One can then compare the dynamics predicted by the ODEs and that obtained from the physical system. However, there are now many large data sets from physical and biological systems for which we know much less about the underlying physical principles. In light of this, an important goal is to be able to generate systems of ODEs that recover the measured time series data and can be used to predict system behavior in parameter regimes or time domains that have not yet been measured.

ODE models are used extensively in computational biology. For example, in systems biology, genetic circuits are modeled as networks of electronic circuit elements [1, 2]. In addition, systems of ODEs are often employed to investigate viral dynamics (e.g. HIV [3, 4, 5], hepatitis [6, 7], and influenza [8, 9]) and the immune system response to infection [10, 11]. Population dynamics and epidemics have also been successfully modeled using systems of ODEs [12]. In some cases, an \textit{ad hoc} ODE model with several parameters is posited [13], and solutions of the model are compared to experimental data to identify the relevant range of parameter values. A more sophisticated approach [14, 15, 16, 17, 18, 19, 20, 21] is to reconstruct systems of ODEs by expanding the ODEs in a particular, \textit{fixed} basis such as polynomials, rational, harmonic, or radial basis functions. The basis expansion coefficients can then be obtained by fitting the model to the time series data.

A recent study has developed a more systematic computational approach to identify the “best” ODE model to recapitulate time series data. The approach iteratively generates random mathematical expressions for a candidate ODE model. At each iteration, the ODE model is solved and the solution is compared to the time series data to identify the parameters in the candidate model. The selected parameters minimize the distance between the input trajectories and the solutions of the candidate model. Candidate models with small errors are then co-evolved using a genetic algorithm to improve the fit to the input time series data [22, 23, 24, 25]. The advantage of this method is that it yields an approximate analytical expression for the ODE model for the dynamical system. The disadvantages of this approach include the computational expense of repeatedly solving ODE models and the difficulty in finding optimal solutions for multi-dimensional nonlinear regression.

\textsuperscript{*}Corresponding Author
Email address: corey.o hern@yale.edu (Corey S. O'Hern)
Here, we develop a new methodology to build numerical expressions of a system of ODEs that can recapitulate time series data of a dynamical system. This method has the advantage of not needing any input except the time series data, although a priori information about the fixed point structure and basins of attraction of the dynamical system would improve reconstruction. Our method includes several steps. We first learn a basis to sparsely represent the time series data using sparse dictionary learning [26, 27, 28]. We then find the sparsest expansion in the learned basis that is consistent with the measured data. This step can be formulated as solving an underdetermined system of linear equations. We will solve the underdetermined systems using L1-norm regularized regression, which finds the solution to the system with the fewest nonzero expansion coefficients in the learned basis.

An advantage of our approach is that we learn a sparse representation (basis and expansion coefficients) of the time series data with no prior assumptions concerning the form of the basis. In contrast, many previous studies have pre-selected a particular set of basis functions to reconstruct a system of ODEs from time series data. In this case, the representation of the data is limited by the forms of the basis functions that are pre-selected by the investigator. In fact, other bases that are not selected may yield better representations.

A key aspect of our work is that we apply two recent advances in machine learning, sparse learning of a basis and sparse function reconstruction from a known basis, to ODE reconstruction. Sparse reconstruction of ODEs from a known basis has been recently investigated [29, 30, 31]. However, a coupling of the two techniques, i.e., learning a basis from incomplete data that is suitable for sparse reconstruction of a system of ODEs, has not yet been implemented.

We test our ODE reconstruction method on time series data generated from known ODE models in one-, two- and three-dimensional systems, including both non-chaotic and chaotic dynamics. We quantify the accuracy of the reconstruction for each system of ODEs as a function of the amount of data used by the method. Further, we solve the reconstructed system of ODEs and compare the solution to the original time series data. The method developed and validated here can now be applied to large data sets for physical and biological systems for which there is no known system of ODEs.

Identifying sparse representations of data (i.e., sparse coding) is well studied. For example, sparse coding has been widely used for data compression, yielding the JPEG, MPEG, and MP3 data formats. Sparse coding relies on the observation that for most signals a basis can be identified for which only a few of the expansion coefficients are nonzero [32, 33, 34]. Sparse representations can provide accurate signal recovery, while at the same time, reduce the amount of information required to define the signal. For example, keeping only the ten largest coefficients out of 64 possible coefficients in an 8×8 two-dimensional discrete cosine basis (JPEG), leads to a size reduction of approximately a factor of 6.

Recent studies have shown that in many cases perfect recovery of a signal is possible from only a small number of measurements of the signal [35, 36, 37, 38, 39]. This work provided a new lower bound for the amount of data required for perfect reconstruction of a signal; for example, in many cases, one can take measurements at frequencies much below the Nyquist sampling rate and still achieve perfect signal recovery. The related field of compressed sensing emphasizes sampling the signal in compressed form to achieve perfect signal reconstruction [37, 38, 39, 40, 41, 42, 43]. Compressed sensing has a wide range of applications from speed up of magnetic resonance image reconstruction [44, 45, 46] to more efficient and higher resolution cameras [47, 48].

Our ODE reconstruction method relies on the assumption that the functions that comprise the “right-hand sides” of the systems of ODEs can be sparsely represented in some basis. A function \( f(x) \) can be sparsely represented by a set of basis functions \( \{\phi_i\}, i = 1, \ldots, n \) if \( f(x) = \sum_{i=1}^{n} c_i \phi_i(x) \) with only a small number \( s \ll n \) of nonzero coefficients \( c_i \). This assumption is not as restrictive as it may seem at first. For example, suppose we sample a two-dimensional function on a discrete 128×128 grid. Since there are \( 128^2 = 16384 \) independent grid points, a complete basis would require at least \( n = 16384 \) basis functions. For most applications, we expect that a much smaller set of basis functions would lead to accurate recovery of the function. In fact, the sparsest representation of the function is the basis that contains the function itself, where only one of the coefficients \( c_i \) is nonzero. Identifying sparse representations of the system of ODEs is also consistent with the physics paradigm of finding the simplest model to explain a dynamical system.

The remainder of this manuscript is organized as follows. In the Materials and Methods section, we provide a formal definition of sets of ODEs and details about obtaining the right-hand side functions of ODEs from numerically differentiating time series data. We then introduce the concept of \( L_1 \)-regularized regression and apply it to the reconstruction of a sparse undersampled signal. We introduce the concept of sparse basis learning to identify a basis in which the ODE can be represented sparsely. At the end of the Materials and Methods section, we define the error metric that we will use to quantify the accuracy of the ODE reconstruction. In the Results section, we perform ODE reconstruction on models in one-, two-, and three-dimensional systems. For each system, we measure the reconstruction accuracy as a function of the amount of data that is used for the reconstruction, showing examples of both accurate and inaccurate reconstructions. We end the manuscript in the Discussion section with a summary and future applications of our method for ODE reconstruction.

2. Materials and Methods

In this section, we first introduce the mathematical expressions that define sets of ordinary differential equations (ODEs). We then describe how we obtain the system of ODEs from time series data. We then present the sparse reconstruction and sparse basis learning methods that we will employ to build sparse representations of the ODE models. We also compare the accuracy of sparse versus non-sparse methods for signal reconstruction. At the end of this section, we introduce the specific
Sparse coding is ideally suited for solving underdetermined systems because it seeks to identify the minimum number of basis functions to represent the signals $f_i$. If we identify a basis that can represent a given set of signals $f_i$ sparsely, an $L_1$-regularized minimization scheme will be able to find the sparsest representation of the signals [49].

2.2. Sparse Coding

In general, the least squares ($L_2$) solution of Eq. 5 possesses many nonzero coefficients $c_i$, whereas the minimal-$L_1$ solution of Eq. 5 is sparse and possesses only a few non-zero coefficients. In the case of underdetermined systems with many available basis functions, it has been shown that a sparse solution obtained via $L_1$ regularization more accurately represents the solution compared to those that are superpositions of many basis functions [49].

A sparse solution to Eq. 5 can be obtained by minimizing the squared differences between the measurements of the signal $\vec{y}$ and the reconstructed signal $\Phi \vec{c}$ subject to a constraint on the $L_1$ norm of $\vec{c}$ [50]:

$$\hat{c} = \arg \min_{\vec{c}} \frac{1}{2} ||\vec{y} - \Phi \vec{c}||^2_2 + \lambda_1 ||\vec{c}||_1,$$  

(6)

where $||\cdot||_p$ denotes the $L_p$ vector norm and $\lambda_1$ is a Lagrange multiplier that penalizes a large $L_1$ norm of $\vec{c}$. The $L_p$ norm of an $n$-dimensional vector $\vec{x}$ is defined as

$$||\vec{x}||_p = \left[\sum_{i=1}^{N} |x_i|^p\right]^{1/p},$$  

(7)

for $p \geq 1$, where $N$ is the number of components of the vector $\vec{x}$. An important point to note is that recent work in sparse coding [36] has shown that the minimal-$L_1$ solution to Eq. 6 is the sparsest. We thus minimize Eq. 6 for each $\lambda_1$ and choose the particular solution with the minimal $L_1$ norm (and a sufficiently small error in the ODE reconstruction). The optimal value of $\lambda_1$ depends on the ODE system and the specific implementation of the ODE reconstruction methodology.

To emphasize the advantages of sparse representations, we now demonstrate how the sparse signal reconstruction method compares to a standard least squares fit. We first construct a sparse signal (with sparsity $s$) in a given basis. We then sample the signal randomly and attempt to recover the signal using the regularized-$L_1$ and least-squares reconstruction methods. We note that for least-squares regression, we choose the solution with the minimal-$L_2$ norm, which is standard for underdetermined least-squares regression algorithms.

For this example, we choose the discrete cosine basis. For a signal size of 100 values, we have a complete and orthonormal basis of 100 functions $\phi_i(i)$ ($n = 0, \ldots, 99$) each with 100 values ($i = 0, \ldots, 99$):

$$\phi_i(i) = F(n) \cos \left[ \frac{\pi}{100} \left( i + \frac{1}{2} \right) n \right],$$  

(8)

where $F(n)$ is a normalization factor

$$F(n) = \begin{cases} \frac{1}{\sqrt{100}} & \text{for } n = 0 \\ \frac{2}{\sqrt{100}} & \text{for } n = 1, \ldots, 99. \end{cases}$$  

(9)
Note that an orthonormal basis is not a prerequisite for the sparse reconstruction method.

Similar to Eq. 3, we can express the signal as a superposition of basis functions,

\[ g(i) = \sum_{j=0}^{99} c_j \phi_j(i). \]

The signal \( \tilde{g} \) of sparsity \( s \) is generated by randomly selecting \( s \) of the coefficients \( c_j \) and assigning them a random amplitude in the range \([-1, 1]\). We then evaluate \( g(i) \) at \( m \) randomly chosen positions \( i \) and attempt to recover \( g(i) \) from the measurements. If \( m < 100 \), recovering the original signal involves solving an underdetermined system of linear equations.

Recovering the full signal \( \tilde{g} \) from a given number of measurements proceeds as follows. After carrying out the \( m \) measurements, we can rewrite Eq. 4 as

\[ \tilde{y} = P \tilde{g}, \]

where \( \tilde{y} \) is the vector of the measurements of \( \tilde{g} \) and \( P \) is the projection matrix with \( m \times n \) entries that are either 0 or 1. Each row has one nonzero element that corresponds to the position of the measurement. For each random selection of measurements of \( \tilde{g} \), we solve the reduced equation

\[ \tilde{y} = \Theta \hat{c}, \]

where \( \Theta = P \Phi \). After solving Eq. 12 for \( \hat{c} \) (with solution \( \hat{c} \)), we obtain a reconstruction of the original signal

\[ \tilde{g}_{\text{rec}} = \Phi \hat{c}. \]

Fig. 1 shows examples of \( L_1 \) and \( L_2 \) reconstruction methods of a signal as a function of the fraction of the total signal \( M = N_m/m \), where \( N_m \) is the number of measurements (out of \( m \)) used for the reconstruction. We studied values of \( M \) in the range 0.2 to 1 for the reconstructions. Even when only a small fraction of the signal is included (down to \( M = 0.2 \)), the \( L_1 \) reconstruction method achieves nearly perfect signal recovery. In contrast, the least-squares method only achieves adequate recovery of the signal for \( M > 0.9 \). Moreover, when only a small fraction of the signal is included, the \( L_2 \) method is dominated by the mean of the measured points and oscillates rapidly about the mean to match each measurement.

In Fig. 2, we measured the recovery error \( d \) between the original \( \tilde{g} \) and recovered \( \tilde{g}_{\text{rec}} \) signals as a function of the fraction \( M \) of the signal included, for several sparsities \( s \) and reconstruction penalties \( \lambda \). We define the recovery error as

\[ d(\tilde{g}, \tilde{g}_{\text{rec}}) = 1 - \frac{\tilde{g} \cdot \tilde{g}_{\text{rec}}}{||\tilde{g}||_2 ||\tilde{g}_{\text{rec}}||_2}, \]

where \( \tilde{g} \cdot \tilde{g}_{\text{rec}} \) denotes the inner product between the two vectors \( \tilde{g} \) and \( \tilde{g}_{\text{rec}} \). This distance function satisfies \( 0 \leq d \leq 2 \), where \( d \geq 1 \) signifies a large difference between \( \tilde{g} \) and \( \tilde{g}_{\text{rec}} \), and \( d = 0 \) indicates \( \tilde{g}_{\text{rec}} = \tilde{g} \).

For sparsity values \( s = 1 \) and 3, we find that there is an optimal regularization parameter value \( \lambda_c = 10^{-4} \). Large values of \( \lambda_c \) imply a large penalty for non-zero reconstruction coefficients, which yields solutions that have mostly, or only coefficients that are zero. For small \( \lambda_c \), the solutions converge towards the least-squares solutions (see Eq. 6). For the optimal \( \lambda_c \), the \( L_1 \)-reconstruction gives small errors \( (d \sim 0) \) for \( M \geq 0.2 \) (Fig. 2). In contrast, the error for the \( L_2 \)-reconstruction method is nonzero for all \( M < 1 \) for all \( s \). For a non-sparse signal \( s = 20 \), the \( L_1 \) and \( L_2 \) reconstruction methods give similar errors for \( M \leq 0.2 \). In this case, the measurements truly undersample the signal, and thus providing less than 20 measurements is not enough to constrain the 20 nonzero coefficients \( c_j \). However, when \( M > 0.2 \), the error from the \( L_1 \)-reconstruction method is less than that from the \( L_2 \) method and is nearly zero for \( M \gtrsim 0.5 \).

2.3. Sparse Basis Learning

The \( L_1 \)-reconstruction method described in the previous section works well if 1) the signal has a sparse representation in some basis and 2) the basis \( \Phi \) (or a subset of it) contains functions similar to the basis in which the signal is sparse. How...
do we proceed with signal reconstruction if we do not know a basis in which the signal is sparse? One method is to use one of the common basis sets, such as wavelets, sines, cosines, or polynomials [51, 52, 53]. Another method is to employ sparse basis learning that identifies a basis in which a signal can be expressed sparsely. This approach is compelling because it does not require significant prior knowledge about the signal, it does not employ pre-selected basis functions, and it allows the basis to be learned even from noisy or incomplete data.

Sparse basis learning seeks to find a basis $\Phi$ that can represent an input of several signals sparsely. We identify $\Phi$ by decomposing the signal matrix $Y = (\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_m)$, where $m$ is the number of signals, into the basis matrix $\Phi$ and coefficient matrix $C$,

$$Y = \Phi C.$$  

(15)

Columns $\tilde{c}_i$ of $C$ are the sparse coefficient vectors that represent the signals $\tilde{y}_i$ in the basis $\Phi$. Both $C$ and $\Phi$ are unknown and can be determined by minimizing the squared differences between the signals and their representations in the basis $\Phi$ subject to the constraint that the coefficient matrix is sparse [54]:

$$\hat{C}, \hat{\Phi} = \arg \min_{C, \Phi} \sum_{i=1}^{m} \left( \frac{1}{2} ||\tilde{y}_i - \Phi \hat{C}_i ||_2^2 + \lambda_1 ||\hat{C}_i ||_1 \right).$$  

(16)

where $\lambda_1$ is a Lagrange multiplier that determines the sparsity of the coefficient matrix $C$. To solve Eq. 16, we employed the open source solver MiniBatchDictionaryLearning, which is part of the python machine learning package scikit-learn [50]. Again, we minimize Eq. 16 for each $\lambda_1$ and choose the particular solution with the minimal $L_1$ norm (and a sufficiently small error in the ODE reconstruction).

To illustrate and provide an example of the basis learning method, we show the results of sparse basis learning on the complex, two-dimensional image of a cat shown in Fig. 3 (a). To learn a sparse basis for this image, we decomposed the original image ($128 \times 128$ pixels) into all possible $8 \times 8$ patches, which totals 14,641 unique patches. The patches were then reshaped into one-dimensional signals $\tilde{y}_i$, each containing 64 values. We chose 100 basis functions, or patches, (columns of $\Phi$) to sparsely represent the input matrix $Y$. Fig. 3 (b) shows the 100 basis functions (patches) that were obtained by solving Eq. 16. The 100x64 matrix $\Phi$ was reshaped into $8 \times 8$ pixel basis functions before plotting. Note that some of the basis functions display complicated features, e.g., lines and ripples of different widths and angles, whereas others are more uniform.

To demonstrate the utility of the sparse basis learning method, we seek to recover the image in Fig. 3 (a) from an undersampled version using the learned basis functions in Fig. 3 (b) and then performing sparse reconstruction (Eq. 6) for each $8 \times 8$ patch of the undersampled image. For this example, we randomly sampled $\approx 30\%$ of the original image. In Fig. 4 (a), the black pixels indicate the random pixels used for the sparse reconstruction of the undersampled image. We decompose the undersampled image into all possible $8 \times 8$ patches, using only the measurements marked by the black pixels in the sampling mask in Fig. 4 (a). While the reconstruction of the image in Fig. 4 (b) is somewhat grainy, this reconstruction method clearly resembles the original image even when it is $70\%$ undersampled.

In this work, we show that one may also use incomplete data to learn a sparse basis. For example, the case of a discrete representation of a two-dimensional system of ODEs is the same problem as basis learning for image reconstruction (Fig. 3). However, learning the basis from solutions of the system of ODEs, does not provide full sampling of the signal (i.e. the right-hand side of the system of ODEs in Eq. 1), because the dynamics of the system is strongly affected by the fixed point structure and the functions are not uniformly sampled.

To learn a basis from incomplete data, we decompose the signal into patches of a given size that depends on the specific problem under consideration. However, there are general guidelines to follow concerning the choice of an optimal patch size. The patch size should be chosen to be sufficiently small so that there are enough patches to create variability in the basis set, but also large enough so that there is at least one measurement in each patch. We will study how the reconstruction error depends on the patch size for each ODE model we consider. After we decompose the signal into patches, we fill in missing data with random numbers. Note that other methods, such as filling in missing data with the mean of the data in a given patch or interpolation techniques, can also be used to obtain similar quality ODE reconstruction.
After filling in missing data, we convert the padded patches (i.e., original plus random signal) into a signal matrix $\hat{Y}$ and learn a basis $\Phi$ to sparsely represent the signal by solving Eq. 16. To recover the signal, we find a sparse representation $\hat{c}$ of the unpadded signal (i.e., without added random values) in the learned basis $\Phi$ by solving Eq. 12, where $P$ is the matrix that selects only the signal entries that have been measured. We then obtain the reconstructed patch by taking the product $\Phi\hat{c}$. We repeat this process for each patch to reconstruct the full domain. For cases in which we obtain different values for the signal at the same location from different patches, we average the result.

2.4. Models

We test our methods for the reconstruction of systems of ODEs using synthetic data, i.e., data generated by numerically solving systems of ODEs, which allows us to test quantitatively the accuracy as a function of the amount of data used in the reconstruction. We present results from systems of ODEs in one, two, and three dimensions with increasing complexity in the dynamics. For an ODE in one dimension (1D), we only need to reconstruct one nonlinear function $f_1$ of one variable $x_1$. In two dimensions (2D), we need to reconstruct two functions ($f_1$ and $f_2$) of two variables ($x_1$ and $x_2$), and in three dimensions (3D), we need to reconstruct three functions ($f_1$, $f_2$, and $f_3$) of three variables ($x_1$, $x_2$, and $x_3$) to reproduce the dynamics of the system. Each of the systems that we study possesses a different fixed point structure in phase space. The 1D model has two stable fixed points and one unstable fixed point, and thus all trajectories evolve toward one of the stable fixed points. The 2D model has one saddle point and one oscillatory fixed point with closed orbits as solutions. The 3D model we study has no stable fixed points and instead possesses chaotic dynamics on a strange attractor.

**1D model.** For 1D, we study the Reynolds model for the immune response to infection [10]:

$$\frac{dx_1}{dt} = f_1(x_1) = k_{pg} x_1 \left( 1 - \frac{x_1}{x_{\infty}} \right) - \frac{k_{pm} s_a x_1}{\mu_m + k_{mp} x_1},$$

(17)

where the pathogen load $x_1$ is unitless, and the other parameters $k_{pg}, k_{pm}, k_{mp}$, and $s_a$ have units of inverse hours. The right-hand side of Eq. 17 is the sum of two terms. The first term enables logistic growth of the pathogen load. In the absence of any other terms, any positive initial value will cause $x_1$ to grow logistically to the steady-state value $x_{\infty}$. The second term mimics a local, non-specific response to an infection, which reduces the pathogen load. For small values of $x_1$, the decrease is proportional to $x_1$. For larger values of $x_1$, the decrease caused by the second term is constant.

We employed the parameter values $k_{pg} = 0.6$, $x_{\infty} = 20$, $k_{pm} = 0.6$, $s_a = 0.005$, $\mu_m = 0.002$, and $k_{mp} = 0.01$, which were used in previous studies of this ODE model [10, 55]. In this parameter regime, Eq. 17 exhibits two stable fixed points at $x_1 = 0$ and 19.49 and one unstable fixed point, separating the two stable fixed points, at $x_1 = 0.31$ (Fig. 5). As shown in Fig. 6, solutions to Eq. 17 with initial conditions $0 \leq x_1 \leq 0.31$ are attracted to the stable fixed point at $x_1 = 0$, while solutions with initial conditions $x_1 > 0.31$ are attracted to the stable fixed point at $x_1 = 19.49$.

**2D model.** In 2D, we focused on the Lotka-Volterra system of ODEs that describe predator-prey dynamics [56]:

$$\frac{dx_1}{dt} = f_1(x_1, x_2) = \alpha x_1 - \beta x_1 x_2,$$

$$\frac{dx_2}{dt} = f_2(x_1, x_2) = -\gamma x_2 + \delta x_1 x_2,$$

(18)

where $x_1$ and $x_2$ describe the prey and predator population sizes, respectively, and are unitless. In this model, prey have a natural growth rate $\alpha$. In the absence of predators, the prey population $x_1$ would grow exponentially with time. With predators present, the prey population decreases at a rate proportional to the product of both the predator and prey populations with a proportionality constant $\beta$ (with units of inverse time). Without predation, the predator population $x_2$ would decrease with death rate $\gamma$. With the presence of prey $x_1$, the predator population grows proportional to the product of the two population sizes $x_1$ and $x_2$ with a proportionality constant $\delta$ (with units of inverse time).
For the Lotka-Volterra system of ODEs, there are two fixed points, one at $x_1 = 0$ and $x_2 = 0$ and one at $x_1 = \gamma / \delta$ and $x_2 = \alpha / \beta$. The stability of the fixed points is determined by the eigenvalues of the Jacobian matrix evaluated at the fixed points. The Jacobian of the Lotka-Volterra system is given by

$$J_{LV} = \begin{pmatrix} \alpha - \beta x_2 & -\beta x_1 \\ \delta x_2 & -\gamma + \delta x_1 \end{pmatrix}. \quad (19)$$

The eigenvalues of the Jacobian $J_{LV}$ at the origin are $\mathcal{J}_1 = \alpha$, $\mathcal{J}_2 = -\gamma$. Since the model is restricted to positive parameters, the fixed point at the origin is a saddle point. The interpretation is that for small populations of predator and prey, the predator population decreases exponentially due to the lack of a food source. While unharmed by the predator, the prey population can grow exponentially, which drives the system away from the zero population state, $x_1 = 0$ and $x_2 = 0$.

The eigenvalues of the Jacobian $J_{LV}$ at the second fixed point $x_1 = \gamma / \delta$ and $x_2 = \alpha / \beta$ are purely imaginary complex conjugates, $\mathcal{J}_1 = -i \sqrt{\alpha \gamma}$ and $\mathcal{J}_2 = i \sqrt{\alpha \gamma}$, where $i^2 = -1$. The purely imaginary fixed point causes trajectories to revolve around it and form closed orbits. The interpretation of this fixed point is that the predators decrease the number of prey, then the predators begin to die due to a lack of food, which in turn allows the prey population to grow. The growing prey population provides an abundant food supply for the predator, which allows the predator to grow faster than the food supply can sustain. The prey population then decreases and the cycle repeats. For the results below, we chose the parameters $\alpha = 0.4$, $\beta = 0.4$, $\gamma = 0.1$, and $\delta = 0.2$ for the Lotka-Volterra system, which locates the oscillatory fixed point at $x_1 = 0.5$ and $x_2 = 1.0$ (Fig. 7).

3D model. In 3D, we focused on the Lorenz system of ODEs [57], which describes fluid motion in a container that is heated from below and cooled from above:

$$\begin{align*}
\frac{dx_1}{dt} &= f_1(x_1, x_2, x_3) = \sigma(x_2 - x_1) \\
\frac{dx_2}{dt} &= f_2(x_1, x_2, x_3) = x_1(x_3 - x_2) - x_2 - \beta x_3, \\
\frac{dx_3}{dt} &= f_3(x_1, x_2, x_3) = x_1 x_2 - \gamma x_3.
\end{align*} \quad (20)$$

where $\sigma, \rho, \beta$ are positive, dimensionless parameters that represent properties of the fluid. In different parameter regimes, the fluid can display quiescent, convective, and chaotic dynamics. The three dimensionless variables $x_1$, $x_2$, and $x_3$ describe the intensity of the convective flow, temperature difference between the ascending and descending fluid, and spatial dependence of the temperature profile, respectively.

The system possesses three fixed points at $(x_1, x_2, x_3) = (0, 0, 0)$, $(-\beta^{1/2}(\rho - 1)^{1/2}, -\beta^{1/2}(\rho - 1)^{1/2}, \rho - 1)$, and $(\beta^{1/2}(\rho - 1)^{1/2}, \beta^{1/2}(\rho - 1)^{1/2}, \rho - 1)$. The Jacobian of the system is given by

$$J_L = \begin{pmatrix} -\sigma & \sigma & 0 \\ \rho - x_3 & -1 & -x_1 \\ x_2 & x_1 & -\beta \end{pmatrix}. \quad (21)$$

When we evaluate the Jacobian (Eq. 21) at the fixed points, we find that each of the three eigenvalues possesses two stable and one unstable eigendirection in the parameter regime $\sigma = 10$, $\rho = 28$ and $\beta = 8 / 3$. With these parameters, the Lorenz system displays chaotic dynamics with Lyapunov exponents $\ell_1 = 0.9$, $\ell_2 = 0.0$, and $\ell_3 = -14.6$ [58]. In Fig. 8, we show the time evolution of two initial conditions in $x_1$-$x_2$-$x_3$ configuration space for this parameter regime.

3. Results

In this section, we present the results of our methodology for ODE reconstruction of data generated from the three systems of
ODEs described in Materials and Methods. As a summary, our ODE reconstruction methodology involves the following eight steps: 1) Choose \( N_i \) random initial conditions, solve the system of ODEs until time \( t = t_{\text{end}} \) for each initial condition, and sample the solutions at times spaced by \( \Delta t \); 2) Calculate numerical derivatives from the trajectories \( x_i(t) \) to obtain the functions \( f_i \); 3) Decompose each \( f_i \) into all possible patches of a given size; 4) Fill in missing values in the patches with random numbers; 5) Learn a basis to sparsely represent all of the patches; 6) Use the learned basis to sparsely reconstruct the functions \( f_i \) based only on the measurements obtained from the \( N_i \) trajectories and then calculate the non-measured values of \( f_i \); 7) Compute the error \( d \) in the reconstruction by comparing all values of the reconstructed functions to all values in the original model, even at locations that were not measured; and 8) Average the error \( d \) over at least 20 reconstructions by repeating steps 1 through 7.

For each system, we measure the error in the reconstruction as a function of the size of the patches used to decompose the signal for basis learning, the sampling time interval \( \Delta \), and the number of trajectories \( N_i \). For each model, we make sure that the total integration time is sufficiently large that the system can reach the stable fixed points or sample the chaotic attractor in the case of the Lorenz system.

### 3.1. Reconstruction of ODEs in 1D

We first focus on the reconstruction of the Reynolds ODE model in 1D (Eq. 17) using time series data. We discretized the domain \( 0 \leq x_1 \leq 19.5 \) using 256 points, \( i = 0, \ldots, 255 \). Because the unstable fixed point at \( x_1 = 0.31 \) is much closer to the stable fixed point at \( x_1 = 0 \) than to the stable fixed point at \( x_1 = 19.49 \), we sampled more frequently in the region \( 0 \leq x_1 \leq 0.6 \) compared to the region \( 0.6 < x_1 \leq 19.5 \). In particular, we uniformly sampled 128 points from the small domain, and uniformly sampled the same number of points from the large domain.

In Fig. 9, we show the error \( d \) (Eq. 14) in recovering the right-hand side of Eq. 17 (\( f_j(x_1) \)) as a function of the size \( p \) of the patches used for basis learning. Each data point in Fig. 9 represents an average over 20 reconstructions using \( N_i = 10 \) trajectories with a sampling time interval \( \Delta t = 1 \). We find that the error \( d \) achieves a minimum below \( 10^{-3} \) in the patch size range \( 30 < p < 50 \). Patch sizes that are too small do not adequately sample \( f_j(x_1) \), while patch sizes that are too large do not include enough variability to select a sufficiently diverse basis set to reconstruct \( f_j(x_1) \). For example, in the extreme case that the patch size is the same size as the signal, we are only able to learn the input data itself, which may be missing data. For the remaining studies of the 1D model, we set \( p = 50 \) as the basis patch size.

In Fig. 10, we show the dependence of the reconstruction error \( d \) on the two \( \lambda \) parameters used for sparse reconstruction, \( \lambda_t \) (Eq. 6), and basis learning, \( \lambda_i \) (Eq. 16), for the 1D Reynolds model. While we find a minimum in the reconstruction error near \( \lambda_t \approx 10^{-4} \), \( d \) is essentially independent of \( \lambda_t \). Thus, we chose \( \lambda_t = \lambda_i = 10^{-4} \) for ODE reconstruction of the 1D Reynolds model. This result indicates that the sparse learning procedure is much less dependent on the regularization than the sparse reconstruction process.

In Fig. 11, we plot the error in the reconstruction of \( f_j(x_1) \) as a function of the sampling time interval \( \Delta t \) for several numbers of trajectories \( N_i = 1, 2, 20, 50, \) and 200. We find that the error decreases with the number of trajectories used in the reconstruction. For \( N_i = 1 \), the error is large with \( d > 0.1 \). For large numbers of trajectories (e.g. \( N_i = 200 \)), the error decreases with decreasing \( \Delta t \), reaching \( d \sim 10^{-5} \) for small \( \Delta t \). The fact that the error in the ODE reconstruction increases with \( \Delta t \) is consistent with notion that the accuracy of the numerical derivative of each trajectory decreases with increasing sampling interval.

We also investigated the reconstruction error as a function of Gaussian noise added to the measurements of derivatives in Fig. 12. We quantify the level of noise by the coefficient of variation \( \nu \). As expected, the reconstruction error \( d \) increases with \( \nu \) for the 1D Reynolds model. In Fig. 13, we show the error in the reconstruction of \( f_j(x_1) \) as a function of the total integration time \( t_{\text{end}} \). We find that \( d \) decreases strongly as \( t_{\text{end}} \) increases for \( t_{\text{end}} < 20 \). For \( t_{\text{end}} > 20 \), \( d \) reaches a plateau value below \( 10^{-4} \), which depends weakly on \( \Delta t \). For characteristic time scales \( t > 20 \), the Reynolds ODE model reaches one of the two stable fixed points, and therefore \( d \) becomes independent of \( t_{\text{end}} \).

In Fig. 14, we compare accurate (using \( N_i = 50 \) and \( \Delta t = 0.1 \)) and inaccurate (using \( N_i = 10 \) and \( \Delta t = 5 \)) reconstructions of \( f_j(i_1) \) for the 1D Reynolds ODE model. Note that we plot \( f_j \) as a function of the scaled variable \( i_1 \). The indexes \( i_1 = 0, \ldots, 127 \) indicate uniformly spaced \( x_1 \) values in the interval \( 0 \leq x_1 \leq 0.6 \), and \( i_1 = 128, \ldots, 256 \) indicate uniformly spaced \( x_1 \) values in the interval \( 0.6 < x_1 \leq 19.5 \).

We find that using large \( \Delta t \) gives rise to inaccurate measurements of the time derivative of \( x_1 \) and, thus, of \( f_j(x_1) \). In addition, large \( \Delta t \) does not allow dense sampling of phase space, especially in regions where the trajectories evolve rapidly. The inaccurate reconstruction in Fig. 14 (b) is even worse than it
seems at first glance. The reconstructed function is identically zero over a wide range of \( i_1 \) (0 \( \leq \) \( i_1 \) \( \leq \) 50) where \( f(i_i) \) is not well sampled, since the default output of a failed reconstruction is zero. It is a coincidence that \( f(i_i) \sim 0 \) in Eq. 17 over the same range of \( i_1 \).

We now numerically solve the reconstructed 1D Reynolds ODE model for different initial conditions and times comparable to \( t_{end} \) and compare these trajectories to those obtained from the original model (Eq. 17). In Fig. 15, we compare the trajectories \( x_f(t) \) for the accurate (\( d \sim 10^{-5} \); Fig. 14 (a)) and inaccurate (\( d \sim 10^{-2} \); Fig. 14 (b)) representations of \( f_i(x_1) \) to the original model for six initial conditions. All of the trajectories for the accurate representation of \( f_i(x_1) \) are nearly indistin-

guishable from the trajectories for the original model, whereas we find large deviations between the original and reconstructed trajectories even at short times for the inaccurate representation of \( f_f(x_1) \).

### 3.2. Reconstruction of Systems of ODEs in 2D

We now investigate the reconstruction accuracy of our method for the Lotka-Volterra system of ODEs in 2D. We find that the results are qualitatively similar to those for the 1D Reynolds ODE model. We map the numerical derivatives for \( N_f \) trajectories with a sampling time interval \( \Delta t \) onto a 128\times128 grid with 0 \( \leq \) \( x_1, x_2 \) \( \leq \) 2. Similar to the 1D model, we find that the error in the reconstruction of \( f_1(x_1, x_2) \) and \( f_2(x_1, x_2) \) possesses a minimum as a function of the patch area used for basis learning, where the location and value at the minimum depends on the parameters used for the reconstruction. For example, for \( N_f = 200, \Delta t = 0.1, \) and averages over 20 independent

Figure 10: Reconstruction error \( d \) for the 1D Reynolds model as a function of the two \( \lambda \) parameters used for (a) sparse reconstruction, \( \lambda_r \) (Eq. 6), and (b) basis learning, \( \lambda_l \) (Eq. 16), for several different numbers of trajectories \( N_r \) = 10 (blue), 20 (green), and 50 (red) used in the reconstruction. The data for each \( N_r \) is averaged over 20 independent reconstructions.

Figure 11: Reconstruction error \( d \) for the 1D Reynolds model as a function of the sampling interval \( \Delta t \) for several different numbers of trajectories \( N_r \) = 1 (circles), 5 (stars), 10 (triangles), 20 (squares), 50 (diamonds), and 200 (pentagons) used in the reconstruction. The data for each \( N_r \) is averaged over 20 independent reconstructions.

Figure 12: Reconstruction error \( d \) for the 1D Reynolds model as a function of the sampling interval \( \Delta t \) for several different values of a coefficient of variation \( \nu \) of Gaussian noise added to the measurements of the derivatives. We show errors for \( \nu = 0 \) (circles), 0.05 (stars), 0.1 (triangles), 0.2 (squares), and 0.5 (diamonds) used in the reconstruction. The data for each \( \nu \) is averaged over 20 independent reconstructions.

Figure 13: Reconstruction error \( d \) for the 1D Reynolds ODE model as a function of the total integration time \( t_{end} \) used for the reconstruction for \( N_f = 20 \) trajectories at several values of the sampling time \( \Delta t = 0.05 \) (circles), 0.1 (stars), and 0.5 (triangles).
Figure 14: Reconstructions (solid blue lines) of $f_1(i_1)$ for the 1D Reynolds ODE model in Eq. 17 using (a) $N_t = 50$ and $\Delta t = 0.1$ and (b) $N_t = 10$ and $\Delta t = 5$. $f_1$ is discretized using 256 points. The indices $i_1 = 0, \ldots, 127$ indicate uniformly spaced $x_1$ values in the interval $0 \leq x_1 \leq 0.6$, and $i_1 = 128, \ldots, 256$ indicate uniformly spaced $x_1$ values in the interval $0.6 < x_1 \leq 19.5$. The exact expression for $f_1(i_1)$ is represented by the open circles.

In Fig. 16, we show the reconstruction error $d$ as a function of the sampling time interval $\Delta t$ for several values of $N_t$ from 5 to 500 trajectories, for a total time $t_{\text{end}}$ that allows several revolutions around the closed orbits, and for patch size $p_{\text{min}}$. As in 1D, we find that increasing $N_t$ decreases the reconstruction error. For $N_t = 5$, $d \sim 10^{-3}$, while $d < 10^{-5}$ for $N_t = 500$. $d$ also decreases with decreasing $\Delta t$, although $d$ reaches a plateau in the small $\Delta t$ limit, which depends on the number of trajectories included in the reconstruction.

In Figs. 17 and 18, we show examples of inaccurate ($d \sim 10^{-2}$) and accurate ($d \sim 10^{-5}$) reconstructions of $f_1(i_1, i_2)$ and $f_2(i_1, i_2)$. The indexes $i_{1,2} = 0, \ldots, 127$ indicate uniformly spaced $x_1$ and $x_2$ values in the interval $0 \leq x_{1,2} \leq 2$. The error reaches a minimum ($d \approx 10^{-5}$) near $p_{\text{min}} \approx 100$, where the regularization parameters have been chosen to be $\lambda_r = \lambda_l = 10^{-4}$. Note that the maximum signal size includes $128^2$ points.

Using the reconstructions of $f_1$ and $f_2$, we solved for the trajectories $x_1(t)$ and $x_2(t)$ (for times comparable to $t_{\text{end}}$) and compared them to the trajectories from the original Lotka-Volterra model (Eq. 18). In Fig. 19 (a) and (b), we show parametric plots ($x_2(t)$ versus $x_1(t)$) for the inaccurate (Fig. 17) and accurate (Fig. 18) reconstructions of $f_1$ and $f_2$, respectively. We
Figure 16: Reconstruction error $d$ for the 2D Lotka-Volterra model (Eq. 18) as a function of the sampling time interval $\Delta t$ for different numbers of trajectories $N_t = 5$ (circles), 10 (stars), 20 (rightward triangles), 50 (squares), 100 (diamonds), 200 (pentagons), and 500 (upward triangles) averaged over 20 independent reconstructions.

Figure 17: Examples of inaccurate reconstructions of $f_1(t_i, t_{i2})$ and $f_2(t_i, t_{i2})$ (with errors $d = 0.04$ and 0.03, respectively) for the 2D Lotka-Volterra system of ODEs (Eq. 18). The indexes $t_{i2} = 0, \ldots, 128$ indicate uniformly spaced $x_1$ and $x_2$ values in the interval $0 \leq x_1, x_2 \leq 2$. Panels (a) and (d) give the exact functions $f_1(t_i, t_{i2})$ and $f_2(t_i, t_{i2})$, panels (b) and (e) give the reconstructions, and panels (c) and (f) indicate the $t_i$-$t_{i2}$ values that were sampled (2% of the 128 $\times$ 128 grid). The white regions in panels (c) and (f) indicate missing data. The color scales are the same as in Fig. 17. This reconstruction was obtained using $N_t = 100$ trajectories, a sampling interval of $\Delta t = 0.01$, and a basis patch area $p^2 = 625$.

Figure 18: Examples of accurate reconstructions of $f_1(t_i, t_{i2})$ and $f_2(t_i, t_{i2})$ (with errors $d = 4\times10^{-5}$ and $5\times10^{-5}$, respectively) for the 2D Lotka-Volterra system of ODEs (Eq. 18). The indexes $t_{i2} = 0, \ldots, 128$ indicate uniformly spaced $x_1$ and $x_2$ values in the interval $0 \leq x_1, x_2 \leq 2$. Panels (a) and (d) give the exact functions $f_1(t_i, t_{i2})$ and $f_2(t_i, t_{i2})$, panels (b) and (e) give the reconstructions, and panels (c) and (f) indicate the $t_i$-$t_{i2}$ values that were sampled (68% of the 128 $\times$ 128 grid). The white regions in panels (c) and (f) indicate missing data. The color scales are the same as in Fig. 17. This reconstruction was obtained using $N_t = 100$ trajectories, a sampling interval of $\Delta t = 0.01$, and a basis patch area $p^2 = 625$.

Figure 19: Parametric plots of the trajectories $x_1(t)$ and $x_2(t)$ (solid lines) for the (a) inaccurate and (b) accurate reconstructions of $f_1$ and $f_2$ in Figs. 17 and 18, respectively, for four different initial conditions indicated by the crosses. The open circles indicate the trajectories from the Lotka-Volterra system of ODEs (Eq. 18).

3.3. Reconstruction of systems of ODEs in 3D

For the Lorenz ODE model, we need to reconstruct three functions of three variables: $f_1(x_1, x_2, x_3)$, $f_2(x_1, x_2, x_3)$, and $f_3(x_1, x_2, x_3)$. Based on the selected parameters $\sigma = 10, \rho = 28$ and $\beta = 8/3$, we chose a $32 \times 32 \times 32$ discretization of the domain $-21 \leq x_1 \leq 21, -29 \leq x_2 \leq 29,$ and $-2 \leq x_3 \leq 50$. We employed patches of size $10 \times 10$ from each of the 32 slices (along $x_3$) of size $32 \times 32$ (in the $x_1$-$x_2$ plane) to perform the basis learning. The choice for the patches to be in the $x_1$-$x_2$ plane is arbitrary; one may also choose patches that extend into 3D. Again, we chose $\lambda_r = \lambda_l = 10^{-4}$.
In Fig. 20, we plot the reconstruction error $d$ versus the sampling time interval $\Delta t$ for several $N_t$ from 500 to $10^4$ trajectories. As found for the 1D and 2D ODE models, the reconstruction error decreases with decreasing $\Delta t$ and increasing $N_t$. $d$ reaches a low-$\Delta t$ plateau that depends on the value of $N_t$. For $N_t = 10^4$, the low-$\Delta t$ plateau value for the reconstruction error approaches $d \sim 10^{-3}$.

Figure 20: Reconstruction error $d$ plotted versus the sampling time interval $\Delta t$ for the 3D Lorenz model Eq. (20) for several different numbers of trajectories $N_t = 500$ (circles), 1000 (stars), 5000 (rightward triangles), and 10000 (squares). Each data point is averaged over 20 independent reconstructions.

In Fig. 21, we visualize the reconstructed functions $f_1$, $f_2$, and $f_3$ for the Lorenz system of ODEs. Panels (a)-(c) represent $f_1$, (d)-(f) represent $f_2$, and (g)-(i) represent $f_3$. The 3D domain is broken into 32 slices (along $x_3$) of $32 \times 32$ grid points in the $x_1$-$x_2$ plane. Panels (a), (d), and (g) give the original functions $f_1$, $f_2$, and $f_3$ in the Lorenz system of ODEs (Eq. 20). Panels (b), (e), and (h) give the reconstructed versions of $f_1$, $f_2$, and $f_3$, and panels (c), (f), and (i) provide the data that was used for the reconstructions (with white regions indicating missing data). The central regions of the functions are recovered with high accuracy. (The edges of the domain were not well-sampled, and thus the reconstruction was not as accurate.) These results show that even for chaotic systems in 3D we are able to achieve accurate ODE reconstruction. In Fig. 22, we compare trajectories from the reconstructed functions to those from the original Lorenz system of ODEs for times comparable to the inverse of the largest Lyapunov exponent. In this case, we find that some of the trajectories from the reconstructed model closely match those from the original model, while others differ from the trajectories of the original model. Since chaotic systems are extremely sensitive to initial conditions, we expect that all trajectories of the reconstructed model will differ from the trajectories of the original model at long times. Despite this, the trajectories from the reconstructed model display chaotic dynamics with similar Lyapunov exponents to those for the Lorenz system of ODEs. In particular, our estimate for the largest Lyapunov exponent for the reconstructed Lorenz model is approximately 0.8, which is obtained by measuring the exponential growth in the separation between two trajectories that possess an initial separation of $10^{-8}$. Thus, we are able to recover the controlling dynamics of the original model. A more thorough discussion of ODE model validation can be found in Ref. [59].

4. Discussion

We developed a method for reconstructing sets of nonlinear ODEs from time series data using machine learning methods involving sparse function reconstruction and sparse basis learning. Using only information from the system trajectories, we first learned a sparse basis, with no a priori knowledge of the underlying functions in the system of ODEs and no pre-selected basis functions, and then reconstructed the system of ODEs in this learned basis. A key feature of our method is its reliance on sparse representations of the system of ODEs. Our results also emphasize that sparse representations provide more accurate reconstructions of systems of ODEs than least-squares approaches.

We tested our ODE reconstruction method on time series data obtained from systems of ODEs in 1D, 2D, and 3D. In 1D, we studied the Reynolds model for the immune response to infection. In the parameter regime we considered, this system possesses only two stable fixed points, and thus all initial conditions converge to these fixed points in the long-time limit. In 2D, we studied the Lotka-Volterra model for predator-prey dynamics. In the parameter regime we studied, this system possesses an oscillatory fixed point with closed orbits. In 3D, we studied the Lorenz model for convective flows. In the parameter regime we considered, the system displays chaotic dynamics on a strange attractor.

For each model, we measured the error in the reconstructed system of ODEs as a function of parameters of the reconstruction method including the sampling time interval $\Delta t$, number of trajectories $N_t$, total time $t_{\text{end}}$ of the trajectory, and size of the patches used for basis function learning. For the 1D model, we also studied the effect of measurement noise on the reconstruction error. In general, the error decreases as more data is used for the reconstruction. We determined the parameter regimes for which we could achieve highly accurate reconstruction with errors $d < 10^{-3}$. We then generated trajectories from the reconstructed systems of ODEs and compared them to the trajectories of the original models. For the 1D model with two stable fixed points, we were able to achieve extremely accurate reconstruction and recapitulation of the trajectories of the original model. Our reconstruction for the 2D model is also accurate and is able to achieve closed orbits for most initial conditions. For some of the initial conditions, smaller sampling time intervals and longer trajectories were needed to achieve reconstructed solutions with closed orbits. In future studies, we will investigate methods to add a constraint that imposes a constant of the motion on the reconstruction method, which will allow us to use larger sampling time intervals and shorter trajectories and still achieve closed orbits. For the 3D chaotic Lorenz system, we can only match the trajectories of the reconstructed and original systems for times that are small compared to the inverse of the largest Lyapunov exponent. Even though the trajectories of the reconstructed and original systems will diverge, we have shown
that the reconstructed and original systems of ODEs possess dynamics with similar Lyapunov exponents. Now that we have validated this ODE reconstruction method on known deterministic systems of ODEs and determined the parameter regimes that yield accurate reconstructions, we will employ this method in future studies to identify new systems of ODEs using time series data from experimental systems for which there is no currently known system of ODEs.

Figure 21: Reconstruction of the Lorenz system of ODEs in 3D (Eq. 20) using the sampling time interval $\Delta t = 10^{-2}$ and $N_t = 10^5$ trajectories. Panels (a)-(c) indicate $f_1(t_1, t_2, t_3)$, (d)-(f) indicate $f_2(t_1, t_2, t_3)$, and (g)-(i) indicate $f_3(t_1, t_2, t_3)$, where the indexes $t_1$, $t_2$, and $t_3$ represent uniformly spaced $x_1$, $x_2$, and $x_3$ values on the intervals $-21 \leq t_1 \leq 21$, $-29 \leq t_2 \leq 29$, and $-2 \leq t_3 \leq 50$. Each of the 32 panels along $x_3$ represents a $32 \times 32$ discretization of the $x_1$-$x_2$ domain. The first rows of each grouping of three (i.e. panels (a), (d), and (g)) give the original functions $f_1$, $f_2$, and $f_3$. The second rows of each grouping of three (i.e. panels (b), (e), and (h)) give the reconstructed versions of $f_1$, $f_2$, and $f_3$. The third rows of each grouping of three (i.e. panels (c), (f), and (i)) show the points in the $x_1$, $x_2$, and $x_3$ domain that were used for the reconstruction. The white regions indicate missing data. The color scales range from dark blue to red corresponding to the ranges of $-500 \leq f_1(t_1, t_2, t_3) \leq 500$, $-659 \leq f_2(t_1, t_2, t_3) \leq 659$, and $-742 \leq f_3(t_1, t_2, t_3) \leq 614$ for the groups of panels (a)-(c), (d)-(f), and (g)-(i), respectively.

Figure 22: We compare the trajectories $x_1(t)$, $x_2(t)$, and $x_3(t)$ from the reconstructed (solid lines) and original (empty circles) functions $f_1$, $f_2$, and $f_3$ from the 3D Lorenz system of ODEs (Eq. 20) with the parameters $\sigma = 10, \rho = 28$, and $\beta = 8/3$ in the chaotic regime. We plot $x_1$, $x_2$, and $x_3$ parametrically for two initial conditions indicated by the crosses.

References


