MULTISCALE MODEL. SIMUL.
Vol. 17, No. 3, pp. 948–972

ON THE CONVERGENCE OF THE SINDy ALGORITHM*

LINAN ZHANG† AND HAYDEN SCHAEFFER†

Abstract. One way to understand time-series data is to identify the underlying dynamical system which generates it. This task can be done by selecting an appropriate model and a set of parameters which best fits the dynamics while providing the simplest representation (i.e., the smallest amount of terms). One such approach is the *sparse identification of nonlinear dynamics* framework [6], which uses a sparsity-promoting algorithm that iterates between a partial least-squares fit and a thresholding (sparsity-promoting) step. In this work, we provide some theoretical results on the behavior and convergence of the algorithm proposed in [S. L. Brunton, J. L. Proctor, and J. N. Kutz, *Proc. Nat. Acad. Sci. USA*, 113 (2016), pp. 3932–3937]. In particular, we prove that the algorithm approximates local minimizers of an unconstrained ℓ^0-penalized least-squares problem. From this, we provide sufficient conditions for general convergence, rate of convergence, conditions for one-step recovery, and a recovery result with respect to the condition number and noise. Examples illustrate that the rates of convergence are sharp. In addition, our results extend to other algorithms related to the algorithm in [S. L. Brunton, J. L. Proctor, and J. N. Kutz, *Proc. Nat. Acad. Sci. USA*, 113 (2016), pp. 3932–3937], and provide theoretical verification of several observed phenomena.

Key words. sparse approximation, convergence analysis, model selection

AMS subject classifications. 65F10, 65K99, 68W40, 90C26

DOI. 10.1137/18M1189828

1. Introduction.

Dynamic model identification arises in a variety of fields where one would like to learn the underlying equations governing the evolution of some given time-series data $u(t)$. This is often done by learning a first-order differential equation $\dot{u} = f(u)$ which provides a reasonable model for the dynamics. The function f is unknown and must be learned from the data. Some applications including weather modeling and prediction, development and design of aircraft, modeling the spread of disease over time, trend predictions, etc.

Several analytical and numerical approaches have been developed to solve various model identification problems. One important contribution to model identification is [4, 26], where the authors introduced a symbolic regression algorithm to determine underlying physical equations, like equations of motion or energies, from data. The key idea is to learn the governing equation directly from the data by fitting the derivatives with candidate functions while balancing between accuracy and parsimony. In [6], the authors proposed the *sparse identification of nonlinear dynamics (SINDy) algorithm*, which computes sparse solutions to linear systems related to model identification and parameter estimation. The main idea is to convert the (nonlinear) model identification problem to a linear system,

$$Ax = b, \tag{1.1}$$

where the matrix $A \in \mathbb{R}^{m\times n}$ is a data-driven dictionary whose columns are (nonlinear) candidate functions of the given data u, the unknown vector $x \in \mathbb{R}^n$ represents the

*Received by the editors May 24, 2018; accepted for publication (in revised form) May 1, 2019; published electronically July 25, 2019.

https://doi.org/10.1137/18M1189828

Funding: The authors were supported by AFOSR grant FA9550-17-1-0125 and NSF CAREER grant 1752116.

†Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213 (linanz@andrew.cmu.edu, schaeffer@cmu.edu).

coefficients of the selected terms in the governing equation f, and the vector $b \in \mathbb{R}^m$ is an approximation to the first-order time derivative $\dot{u}$. In this method, the number of candidate functions are fixed, and thus one assumes that the set of candidate functions is sufficiently large to capture the nonlinear dynamics present in the data. In practice, (1.1) defines an overdetermined linear system, i.e., $m \geq n$. In order to select an accurate model (from the set of candidate functions) which does not overfit the data, the authors of [6] proposed a sparsity-promoting algorithm. In particular, a sparse vector x which approximately solves (1.1) is generated by the following iterative scheme:

$$S^k = \{1 \leq j \leq n : |x_j^k| \geq \lambda\}, \tag{1.2a}$$

$$x^{k+1} = \underset{x \in \mathbb{R}^n:\ \mathrm{supp}(x) \subseteq S^k}{\operatorname{argmin}} \|Ax - b\|_2, \tag{1.2b}$$

where $\lambda > 0$ is a thresholding parameter, and $\mathrm{supp}(x)$ is the support set of x. In practice, it was observed that the algorithm converged within a few steps and produced an appropriate sparse approximation to (1.1). These observations are quantified in our work.

There are several approaches which leverage sparse approximations for model identification. In [10], the authors combined the SINDy framework with model predictive control to solve model identification problems given noisy data. The resulting algorithm is able to control nonlinear systems and identify models in real-time. In [15], the authors introduced information criteria to the SINDy framework, where they selected the optimal model (with respect to the chosen information criteria) over various values of the thresholding parameter. Other approaches have been developed based on the SINDy framework, including SINDy for rational governing equations [14], SINDy with control [7], and SINDy for abrupt changes [18].

In [22], a sparse regression approach for identifying dynamical systems via the weak form was proposed. The authors used the following constrained minimization problem:

$$\min_x \|x\|_0 \quad \text{subject to } \|Ax - b\|_2 \leq \sigma,$$

where the dictionary matrix A is formulated using an integrated set of candidate functions. In [23], several sampling strategies were developed for learning dynamical equations from high-dimensional data. It was proven analytically and verified numerically that, under certain conditions, the underlying equation can be recovered exactly from the following constrained minimization problem, even when the data is undersampled:

$$\min_x \|x\|_1 \quad \text{subject to } \|Ax - b\|_2 \leq \sigma,$$

where the dictionary matrix A consists of second-order Legendre polynomials applied to the data. In [24], the authors developed an algorithm for learning dynamics from multiple time-series data whose governing equations have the same form but different (unknown) parameters. The authors provided convergence guarantees for their group-sparse hard thresholding pursuit algorithm; in particular, one can recover the dynamics when the data-drive dictionary is coercive. The linear system considered in [30] is $AX = B$, where $X \in \mathbb{R}^{n \times d}$ is unknown, and is associated with the first-order differential equation $\dot{\mathbf{u}} = \mathbf{f}(\mathbf{u})$ in $\mathbb{R}^d$. To separate the corrupted data from the

uncorrupt points, the authors solved the minimization problem[1]

$$\min_{x,\eta} \|\eta\|_{2,1} \quad \text{subject to } AX + \eta = B \ \text{ and } \ x \text{ is sparse},$$

where the residual $\eta := B - AX$ is the variable representing the (unknown) corrupted locations. It was shown in [30, 9] that the linear system is well conditioned (when n is large enough) if the data is sampled from a chaotic process, even when the measurements are dependent. Therefore, these learning approaches are appropriate for multiscale and chaotic dynamics. When the data is a function of both time and space, i.e., $u = u(t, y)$ for some spatial variable y, the dictionary can incorporate spatial derivatives [21, 20]. In [21], an unconstrained ℓ^1-regularized least-squares problem (LASSO [29]) with a dictionary built from nonlinear functions of the data and its spatial partial derivatives was used to discover partial differential equations (PDEs) from data. In [20], the authors proposed an adaptive SINDy algorithm for discovering PDEs, which iteratively applies ridge regression with hard thresholding. Additional approaches for model identification can be found in [27, 5, 8, 12, 16, 11, 19, 25].

The sparse model identification approaches include a sparsity-promoting substep, typically through various thresholding operations. In particular, the SINDy algorithm, i.e., (1.2), alternates between a reduced least-squares problem and a thresholding step. This is related to, but differs from, the iterative thresholding methods widely used in compressive sensing. To find a sparse representation of x in (1.1), it is natural to solve the ℓ^0-minimization problems, where the ℓ^0-penalty of a vector measures the number of its nonzero elements. In [3, 1, 2], the authors provided iterative schemes to solve the unconstrained and constrained ℓ^0-regularized problems:

$$\min_x \|Ax - b\|_2^2 + \lambda^2 \|x\|_0, \tag{1.3}$$

$$\min_x \|Ax - b\|_2^2 \quad \text{subject to } \|x\|_0 \le s, \tag{1.4}$$

respectively, where $\|A\|_2 = 1$. To solve (1.3), one iterates

$$x^{k+1} = H_\lambda(x^k + A^T(b - Ax^k)), \tag{1.5}$$

where H_λ is the hard thresholding operator defined componentwise by

$$H_\lambda(x)_j := \operatorname{sgn}(x_j) \max(|x_j|, \lambda).$$

To solve (1.4), one iterates

$$x^{k+1} = L_s(x^k + A^T(b - Ax^k)), \tag{1.6}$$

where L_s is a nonlinear operator that only retains s elements of x with the largest magnitude and sets the remaining $n - s$ elements to zero.

The authors of [3, 1, 2] also proved that the iterative algorithms defined by (1.5) and (1.6) converge to the local minimizers of (1.3) and (1.4), respectively, and derived theoretically the error bounds and convergence rates for the solutions obtained via (1.5). In this work, we will show that the SINDy algorithm also finds local minimizers of (1.3) and has similar theoretical guarantees.

[1] The $\ell^{2,1}$ norm of a matrix $M \in \mathbb{R}^{n\times m}$ is defined by $\|M\|_{2,1} := \sum_{i=1}^n \|M_{i,:}\|_2$.

1.1. Contribution. In this work, we show (in section 2) that the SINDy algorithm proposed in [6] approximates the local minimizers of (1.3), an unconstrained and nonconvex objective function. We provide sufficient conditions for convergence and bounds on rate of convergence. We also prove that the algorithm typically converges to a local minimizer rapidly (in a finite number of steps). Based on several examples, the rate of convergence is sharp. We also show that the convergence results can be adapted to other SINDy-based algorithms. In section 3, we highlight some of the theoretical results by applying the algorithm from [6] to identify dynamical systems from noisy measurements.

2. Convergence analysis. Before detailing the results, we briefly introduce some notations and conventions. For an integer $n \in \mathbb{N}$, let $[n] \subset \mathbb{N}$ be the set defined by $[n] := \{1, 2, \dots, n\}$. Let A be a matrix in $\mathbb{R}^{m\times n}$, where $m \geq n$. If A is injective (or equivalently if A is full column rank, i.e., $\operatorname{rank}(A) = n$), then its pseudo-inverse $A^\dagger \in \mathbb{R}^{n\times m}$ is defined as $A^\dagger := (A^T A)^{-1} A^T$. Let $x \in \mathbb{R}^n$, and define the support set of x as the set of indices corresponding to its nonzero elements, i.e.,

$$\operatorname{supp}(x) := \{j \in [n] : x_j \neq 0\}.$$

The ℓ^0 penalty of x measures the number of nonzero elements in the vector and is defined as

$$\|x\|_0 := \operatorname{card}(\operatorname{supp}(x)).$$

The vector x is called s-sparse if it has at most s nonzero elements; thus $\|x\|_0 \leq s$.

Given a set $S \subseteq [n]$, where $n \in \mathbb{N}$ is known from the context, define $\bar{S} := [n]\backslash S$. For a matrix $A \in \mathbb{R}^{m\times n}$ and a set $S \subseteq [n]$, we denote by A_S the submatrix of A in $\mathbb{R}^{m\times s}$ which consists of the columns of A with indices $j \in S$, where $s = \operatorname{card}(S)$. Similarly, for a vector $x = \left(x_1, x_2, \dots, x_n\right)^T$, let x_S be the subvector of x in $\mathbb{R}^s$ consisting of the elements of x with indices $j \in S$, or the vector in $\mathbb{R}^n$ which coincides with x on S and is zero outside S:

$$(x_S)_j = \begin{cases} x_j & \text{if } j \in S, \\ 0 & \text{if } j \in \bar{S}. \end{cases}$$

The representation of x_S should be clear within the context.

2.1. Algorithmic convergence. Let $A \in \mathbb{R}^{m\times n}$ be a matrix with $m \geq n$, let $\operatorname{rank}(A) = n$, $x \in \mathbb{R}^n$ be the unknown signal, and let $b \in \mathbb{R}^m$ be the observed data. The results presented here work for general A satisfying these assumptions, but the specific application of interest is detailed in section 3. The SINDy algorithm from [6] is

$$x^0 = A^\dagger b, \tag{2.1a}$$

$$S^k = \{j \in [n] : |x_j^k| \geq \lambda\}, \quad k \geq 0, \tag{2.1b}$$

$$x^{k+1} = \operatorname*{argmin}_{x\in\mathbb{R}^n : \operatorname{supp}(x)\subseteq S^k} \|Ax - b\|_2, \quad k \geq 0, \tag{2.1c}$$

which is used to find a sparse approximation to the solution of $Ax = b$, where $\lambda > 0$ is the free parameter. The following theorem shows that the SINDy algorithm terminates in finite steps.

THEOREM 2.1. *The iterative scheme defined by* (2.1) *converges in at most n steps.*

Proof. Let x^k be the sequence generated by (2.1). By (2.1c) we have $\text{supp}(x^{k+1}) \subseteq S^k$, and from (2.1b) we have $S^{k+1} \subseteq \text{supp}(x^{k+1})$. Therefore, the sets S^k are nested:

$$S^{k+1} \subseteq \text{supp}(x^{k+1}) \subseteq S^k. \tag{2.2}$$

Consider the following two cases. If there exists an integer $M \in \mathbb{N}$ such that $S^{M+1} = S^M$, then

$$x^{M+2} = \underset{\text{supp}(x)\subseteq S^{M+1}}{\text{argmin}} \|Ax - b\|_2 = \underset{\text{supp}(x)\subseteq S^{M}}{\text{argmin}} \|Ax - b\|_2 = x^{M+1}. \tag{2.3}$$

Thus, $x^k = x^{M+1}$ for all $k \geq M+1$, and $S^k = S^M$ for all $k \geq M$. Since $\text{card}(S^k) \leq n$ for all $k \in \mathbb{N}$, we conclude that $M \leq n$, so that the scheme converges in at most n steps.

On the other hand, if there does not exist an integer M such that $S^{M+1} = S^M$ and $S^M \neq \emptyset$, then we have a sequence of strictly nested sets, i.e.,

$$S^{k+1} \subsetneq S^k \quad \text{for all } k \text{ such that } S^k \neq \emptyset.$$

Since $\text{card}(S^k) \leq n$ for all $k \in \mathbb{N}$, we must have $S^k = \emptyset$ for all $k > n$. Therefore, the scheme converges to the trivial solution within n steps. ∎

Remark 2.2. Equation (2.3) suggests that an appropriate stopping criterion for the scheme is that the sets are stationary, i.e., $S^k = S^{k-1}$.

Note that, since the support sets are nested, the scheme will converge in at most $\text{card}(S^0)$ steps. The following is an immediate consequence of Theorem 2.1.

COROLLARY 2.3. *The iterative scheme defined by* (2.1) *converges to an s-sparse solution in at most* $\text{card}(S^0) - s$ *steps.*

2.2. Convergence to the local minimizers. In this section, we will show that the iterative scheme defined by (2.1) produces a minimizing sequence for a nonconvex objective associated with sparse approximations. This will lead to a clearer characterization of the fixed-points of the iterative scheme.

Without loss of generality, assume in addition that $\|A\|_2 = 1$. We first show that the scheme converges to a local minimizer of the following (nonconvex) objective function:

$$F(x) := \|Ax - b\|_2^2 + \lambda^2 \|x\|_0, \quad x \in \mathbb{R}^n. \tag{2.4}$$

THEOREM 2.4. *The iterates* x^k *generated by* (2.1) *strictly decrease the objective function unless the iterates are stationary.*

Proof. Define the auxiliary variable:

$$y^k := x^k_{S^k}, \quad k \in \mathbb{N}, \tag{2.5}$$

which plays the role of an intermediate approximation. In particular, we will relate x^{k+1} and y^k.

Observe that (2.1) emits several useful properties. First, we have shown in (2.2) that $S^{k+1} \subseteq \text{supp}(x^{k+1}) \subseteq S^k$. Next, by (2.1c), x^{k+1} is the least-squares solution over the set S^k. By considering the derivative of $\|Ax - b\|_2^2$ with respect to x, we obtain that

$$\left(A^T(Ax^{k+1} - b)\right)_{S^k} = 0, \tag{2.6}$$

and the solution x^{k+1} to the above equation satisfies $x^{k+1}_{S^k} = (A_{S^k})^\dagger b$. To relate x^{k+1} and y^k, note that (2.2) and (2.5) imply that

$$\operatorname{supp}(x^{k+1}) \subseteq \operatorname{supp}(y^k) = S^k \subseteq \operatorname{supp}(x^k). \tag{2.7}$$

By (2.1c) and (2.7), we have

$$\|Ax^{k+1} - b\|_2 \leq \|Ay^k - b\|_2 \quad \text{and} \quad \|x^{k+1}\|_0 \leq \|y^k\|_0, \tag{2.8}$$

respectively.

To show that the objective function decreases, we use the optimization transfer technique as in [3]. Define the surrogate function G for F:

$$G(x,y) := \|Ax - b\|_2^2 - \|A(x-y)\|_2^2 + \|x-y\|_2^2 + \lambda^2\|x\|_0, \quad x \in \mathbb{R}^n. \tag{2.9}$$

Since $\|A\|_2 = 1$, the term $-\|A(x-y)\|_2^2 + \|x-y\|_2^2$ is nonnegative:

$$-\|A(x-y)\|_2^2 + \|x-y\|_2^2 \geq -\|A\|_2^2\|x-y\|_2^2 + \|x-y\|_2^2 = 0,$$

and thus we have $G(x,y) \geq F(x)$ and $G(x,x) = F(x)$ for all $x, y \in \mathbb{R}^n$.

Define the matrix $B := I - A^TA$. Since A is injective (which is implied by $\operatorname{rank}(A) = n$) and $\|A\|_2 = 1$, we have that the eigenvalues of B are in the interval $[0,1]$. Fixing the index $k \in \mathbb{N}$, from (2.4) and (2.8)–(2.9), we have

$$\begin{aligned} F(x^{k+1}) &= \|Ax^{k+1} - b\|_2^2 + \lambda^2\|x^{k+1}\|_0 \\ &\leq \|Ax^{k+1} - b\|_2^2 + \lambda^2\|x^{k+1}\|_0 + \|x^k - y^k\|_B^2 \\ &\leq \|Ay^k - b\|_2^2 + \lambda^2\|y^k\|_0 + \|x^k - y^k\|_B^2 = G(y^k, x^k). \end{aligned}$$

It remains to show that $G(y^k, x^k) \leq G(x^k, x^k)$. By (2.5), we have

$$x^k - y^k = x^k - x^k_{S^k} = x^k_{\operatorname{supp}(x^k)} - x^k_{S^k \cap \operatorname{supp}(x^k)} = x^k_{\bar{S}^k \cap \operatorname{supp}(x^k)},$$

where we included the intersection with $\operatorname{supp}(x^k)$ to emphasize that the difference is zero outside of the support set of x^k. Thus, the difference with respect to the surrogate function simplifies to

$$\begin{aligned} &G(y^k, x^k) - G(x^k, x^k) \\ &\quad = \|Ay^k - b\|_2^2 - \|A(y^k - x^k)\|_2^2 + \|y^k - x^k\|_2^2 + \lambda^2\|y^k\|_0 - \|Ax^k - b\|_2^2 - \lambda^2\|x^k\|_0 \\ &\quad = -2\langle b, Ay^k\rangle + 2\langle Ay^k, Ax^k\rangle - 2\|Ax^k\|_2^2 + 2\langle b, Ax^k\rangle + \|x^k - y^k\|_2^2 \\ &\quad + \lambda^2\left(\|y^k\|_0 - \|x^k\|_0\right) \\ &\quad = -2\langle y^k - x^k, A^T(b - Ax^k)\rangle + \|x^k - y^k\|_2^2 + \lambda^2\left(\|y^k\|_0 - \|x^k\|_0\right) \\ &\quad = -2\langle x^k_{\bar{S}^k \cap \operatorname{supp}(x^k)}, A^T(Ax^k - b)\rangle + \|x^k_{\bar{S}^k \cap \operatorname{supp}(x^k)}\|_2^2 + \lambda^2\left(\|y^k\|_0 - \|x^k\|_0\right). \end{aligned} \tag{2.10}$$

By (2.2) and (2.6), we can observe that

$$\operatorname{supp}(x^k_{\bar{S}^k \cap \operatorname{supp}(x^k)}) \subseteq \operatorname{supp}(x^k) \subseteq S^{k-1} \quad \text{and} \quad (A^T(Ax^k - b))_{S^{k-1}} = 0,$$

respectively, which together imply that

$$\langle x^k_{\bar{S}^k \cap \mathrm{supp}(x^k)}, A^T(Ax^k - b)\rangle = 0. \tag{2.11}$$

In addition, by (2.7), we have

$$\mathrm{card}(S^k) - \mathrm{card}(\mathrm{supp}(x^k)) = -\mathrm{card}(\mathrm{supp}(x^k)\backslash S^k) = -\mathrm{card}(\bar{S}^k \cap \mathrm{supp}(x^k)). \tag{2.12}$$

Consider the following two cases. If $\bar{S}^k \cap \mathrm{supp}(x^k) = \emptyset$, then by (2.2), we must have $\mathrm{supp}(x^k) = S^k$, i.e., $x^k = x^{k+1}$. Therefore, x^k is a fixed point, and $F(x^\ell)$ is stationary for all $\ell \geq k$. If $\bar{S}^k \cap \mathrm{supp}(x^k) \neq \emptyset$, then there exists an integer $j \in \bar{S}^k \cap \mathrm{supp}(x^k)$ such that $|x^k_j| < \lambda$, and thus

$$\|x^k_{\bar{S}^k \cap \mathrm{supp}(x^k)}\|_2^2 < \lambda^2 \,\mathrm{card}(\bar{S}^k \cap \mathrm{supp}(x^k)). \tag{2.13}$$

Thus, by (2.12) and (2.13), provided that the iterates are not stationary, we have

$$\begin{aligned} &\|x^k_{\bar{S}^k \cap \mathrm{supp}(x^k)}\|_2^2 + \lambda^2 \left(\|y^k\|_0 - \|x^k\|_0\right) \\ &\quad = \|x^k_{\bar{S}^k \cap \mathrm{supp}(x^k)}\|_2^2 + \lambda^2 \left(\mathrm{card}(S^k) - \mathrm{card}(\mathrm{supp}(x^k))\right) \\ &\quad < \lambda^2 \mathrm{card}(\bar{S}^k \cap \mathrm{supp}(x^k)) + \lambda^2 \left(\mathrm{card}(S^k) - \mathrm{card}(\mathrm{supp}(x^k))\right) = 0. \end{aligned} \tag{2.14}$$

Combining (2), (2.11), and (2.14) yields

$$G(y^k, x^k) - G(x^k, x^k) < 0.$$

Therefore, for $k \in \mathbb{N}$ such that the k-iteration is not stationary, we have

$$F(x^{k+1}) \leq G(y^k, x^k) < G(x^k, x^k) = F(x^k),$$

which completes the proof. ∎

In the following theorem, we show that the scheme converges to a fixed point which is a local minimizer of the objective function F.

THEOREM 2.5. *The iterates x^k generated by* (2.1) *converge to a fixed point of the iterative scheme defined by* (2.1). *A fixed point of the scheme is also a local minimizer of the objective function defined by* (2.4).

Proof. We first observe that

$$\|Ax^k - b\|_2 \leq \|b\|_2 \tag{2.15}$$

for all $k \in \mathbb{N}$. This is an immediate consequence of (2.1c),

$$\|Ax^k - b\|_2 = \min_{x \in \mathbb{R}^n : \mathrm{supp}(x) \subseteq S^{k-1}} \|Ax - b\|_2 \leq \|b\|_2,$$

since the zero vector is in the feasible set. Next, we show that

$$\sum_{k=1}^{\infty} \|x^{k+1} - x^k\|_2^2 < \infty. \tag{2.16}$$

Denote the smallest eigenvalue of A^TA by λ_0. The assumption that A has full column rank implies that $\lambda_0 > 0$. Thus, by the coercivity of A^TA,

$$\|x^{k+1} - x^k\|_2^2 \leq \frac{1}{\lambda_0} \|A(x^{k+1} - x^k)\|_2^2 \tag{2.17}$$

for all $k \in \mathbb{N}$. For $k \in \mathbb{N}$, define subsets $W^k, V^k \subseteq \mathbb{R}^m$ by

$$W^k := \{Ax : x \in \mathbb{R}^n, \ \ \mathrm{supp}(x) \subseteq S^k\},$$
$$V^k := \{r \in \mathbb{R}^m : \langle r, y\rangle = 0 \ \ \forall y \in W^k\} = (W^k)^\perp.$$

Fixing $M \in \mathbb{N}$ and $k \in [M]$ and setting $r := b - Ax^k$, we have $(A^T r)_{S^{k-1}} = 0$ by (2.6). For $x \in \mathbb{R}^n$ with $\mathrm{supp}(x) \subseteq S^{k-1}$, we have $\langle r, Ax\rangle = \langle A^T r, x\rangle = 0$, which implies that $r \in V^{k-1}$. In addition, $A(x - x^k) \in W^{k-1}$ for all $x \in \mathbb{R}^n$ with $\mathrm{supp}(x) \subseteq S^{k-1}$. Thus,

$$\begin{aligned} \|Ax - b\|_2^2 &= \|A(x - x^k)\|_2^2 - 2\langle A(x - x^k), r\rangle + \|Ax^k - b\|_2^2 \\ &= \|A(x - x^k)\|_2^2 + \|Ax^k - b\|_2^2 \end{aligned} \tag{2.18}$$

for all $x \in \mathbb{R}^n$ with $\mathrm{supp}(x) \subseteq S^{k-1}$. By (2.2) and (2.18),

$$\|A(x^{k+1} - x^k)\|_2^2 = \|Ax^{k+1} - b\|_2^2 - \|Ax^k - b\|_2^2 \tag{2.19}$$

for all $k \in [M]$. Combining (2.15), (2.17), and (2.19) yields

$$\begin{aligned} \sum_{k=1}^{M} \|x^{k+1} - x^k\|_2^2 &\le \frac{1}{\lambda_0} \sum_{k=1}^{M} \|A(x^{k+1} - x^k)\|_2^2 \\ &= \frac{1}{\lambda_0} \sum_{k=1}^{M} \left(\|Ax^{k+1} - b\|_2^2 - \|Ax^k - b\|_2^2\right) \\ &\le \frac{1}{\lambda_0} \ \|Ax^{M+1} - b\|_2^2 \le \frac{1}{\lambda_0}\|b\|_2^2, \end{aligned}$$

and (2.16) follows by sending $M \to \infty$.

We now show that the iterates x^k converge to a fixed point of the scheme. Since $\|x^{k+1} - x^k\|_2 \to 0$ as $k \to \infty$, for any $\epsilon > 0$, there exists an integer $N \in \mathbb{N}$ such that $\|x^{k+1} - x^k\| < \epsilon$ for all $k \ge N$. Assume to the contrary that the scheme does not converge. Then there exists an integer $K \ge N$ such that $S^K \backslash S^{K+1} \neq \emptyset$. Thus, we can find an index $j \in S^K \backslash S^{K+1}$. By (2.1), we must have $|x_j^K| \ge \lambda$, $|x_j^{K+1}| < \lambda$, and $x_j^{K+2} = 0$. Thus,

$$\lambda - |x_j^{K+1}| \le |x_j^K - x_j^{K+1}| \le \|x^{K+1} - x^K\|_2 < \epsilon$$

and

$$|x_j^{K+1}| = |x_j^{K+1} - x_j^{K+2}| \le \|x^{K+2} - x^{K+1}\|_2 < \epsilon.$$

The two conditions on $|x_j^{K+1}|$ above imply that $\lambda - \epsilon < |x_j^{K+1}| < \epsilon$, which fails when, for example, $\epsilon = \lambda/3$. Therefore, the iterates x^k converge. In particular, the preceding argument indicates that there exists an integer $N \in \mathbb{N}$ such that $S^k \backslash S^{k+1} = \emptyset$ for all $k \ge N$. Since the sets S^k are nested, we conclude that $S^{k+1} = S^k$ for all $k \ge N$. Therefore, the iterates x^k converge to a fixed point of the scheme defined by (2.1).

We now show that a fixed point of the scheme is a local minimizer of the objective function defined by (2.4). Let x^* be a fixed point of the scheme. Then x^* and the set $S^* := \mathrm{supp}(x^*)$ satisfy

$$S^* = \{j \in [n] : |x_j^*| \ge \lambda\} \quad \text{and} \quad x^* = \underset{\mathrm{supp}(x) \subseteq S^*}{\mathrm{argmin}} \|Ax - b\|_2. \tag{2.20}$$

From (2.20), we observe that

$$(A^T(Ax^* - b))_{S^*} = 0 \tag{2.21}$$

and

$$x_j^* \neq 0 \iff |x_j^*| \geq \lambda. \tag{2.22}$$

To show that x^* is a local minimizer of F, we will find a positive real number $\epsilon > 0$ such that

$$F(x^* + z) \geq F(x) \quad \text{for all } z \in \mathbb{R}^n \text{ with } \|z\|_\infty < \epsilon. \tag{2.23}$$

Let $U \subseteq [n]$ be the complement of the support set of x^*:

$$U := \{j \in [n] : x_j^* = 0\}.$$

Then by (2.22),

$$\bar{U} = \operatorname{supp}(x^*) = \{j \in [n] : x_j^* \neq 0\} = \{j \in [n] : |x_j^*| \geq \lambda\} = S^*. \tag{2.24}$$

Fixing $z \in \mathbb{R}^n$, from (2.9), we have

$$G(x^* + z, x^*) - G(x^*, x^*) = 2\langle Az, Ax^* - b\rangle + \lambda^2 \left(\|x^* + z\|_0 - \|x^*\|_0\right) + \|z\|_2^2.$$

Let a_j be the jth column of A. Then

$$\begin{aligned} &2\langle Az, Ax^* - b\rangle + \lambda^2 \left(\|x^* + z\|_0 - \|x^*\|_0\right) \\ &\quad = \sum_{j\in U} \left(2a_j^T (Ax^* - b)\, z_j + \lambda^2 |z_j|^0\right) \\ &\qquad + \sum_{j\in \bar{U}} \left(2a_j^T (Ax^* - b)\, z_j + \lambda^2 (|x_j^* + z_j|^0 - |x_j^*|^0)\right) \\ &\quad = \sum_{j\in U} \left(2a_j^T (Ax^* - b)\, z_j + \lambda^2 |z_j|^0\right) + \sum_{j\in \bar{U}} \lambda^2 (|x_j^* + z_j|^0 - |x_j^*|^0), \end{aligned} \tag{2.25}$$

where the last step follows from (2.21). To find an $\epsilon > 0$ such that (2.23) holds, we will show that (3) is nonnegative (so that the difference in G is bounded below by $\|z\|_2^2$).

For $j \in \bar{U}$, we have $|x_j^*| \geq \lambda$ by (2.24). If $|z_j| < \lambda$ for $j \in \bar{U}$, then $x_j^* + z_j \neq 0$, and thus $|x_j^* + z_j|^0 - |x_j^*|^0 = 0$. Therefore, provided that $|z_j| < \lambda$ for all $j \in \bar{U}$,

$$2\langle Az, Ax^* - b\rangle + \lambda^2 \left(\|x^* + z\|_0 - \|x^*\|_0\right) = \sum_{j\in U} \left(2a_j^T (Ax^* - b) z_j + \lambda^2 |z_j|^0\right).$$

For $j \in U$, consider the following two cases. If $z_j = 0$, then the term in the sum is zero:

$$2a_j^T (Ax^* - b) z_j + \lambda^2 |z_j|^0 = 0.$$

If $|z_j| > 0$ and $\lambda^2 \geq 2|a_j^T (Ax^* - b) z_j|$, then

$$2a_j^T (Ax^* - b) z_j + \lambda^2 |z_j|^0 = 2a_j^T (Ax^* - b) z_j + \lambda^2 \geq 0.$$

Combining these results, if ϵ satisfies

$$0 < \epsilon \leq \lambda^2 \min \left\{ \min_{j \in [n]} \frac{1}{2|a_j^T (Ax^* - b)|}, \quad 1 \right\},$$

then for any $z \in \mathbb{R}^n$ with $\|z\|_\infty < \epsilon$, we have

$$G(x^* + z, x^*) - G(x^*, x^*) \geq \|z\|_2^2,$$

which then implies that

$$\begin{aligned} F(x^* + z) &= G(x^* + z, x^*) + \|Az\|_2^2 - \|z\|_2^2 \geq G(x^* + z, x^*) - \|z\|_2^2 \geq G(x^*, x^*) \\ &= F(x^*), \end{aligned}$$

and the proof is complete. ∎

We state a sufficient condition for global minimizers of the objective function in the following theorem.

THEOREM 2.6 (see Theorem 12 from [31]). *Let x^g be a global minimizer of the objective function. Define $U_g := \{j \in [n] : x_j^g = 0\}$. Then,*

$$|a_j^T (Ax^g - b)| \leq \lambda \quad \text{for all } j \in U_g, \tag{2.26a}$$

$$|x_j^g| \geq \lambda \text{ and } a_j^T (Ax^g - b) = 0 \quad \text{for all } j \in \bar{U}_g, \tag{2.26b}$$

where a_j is the jth column of A.

Theorems 2.5 and 2.6 immediately imply the following result.

COROLLARY 2.7. *A global minimizer of the objective function defined by* (2.4) *is a fixed point of the iterative scheme defined by* (2.1).

Theorem 2.5 shows that the iterative scheme converges to a local minimizer of the objective function, but it does not imply that the iterative scheme can obtain all local minima. However, by Corollary 2.7, the global minimizer is indeed obtainable. The following proposition provides a necessary and sufficient condition by which the scheme terminates in one step, which is a consequence of Corollary 2.7.

PROPOSITION 2.8. *Let $x^* \in \mathbb{R}^n$ be a vector which satisfies $Ax^* = b$ and $|x_j^*| \geq \lambda$ on $S := \operatorname{supp}(x^*)$. A necessary and sufficient condition by which x^* can be recovered using the iterative scheme defined by* (2.1) *in one step is*

$$\min_{j \in S} |(A^\dagger b)_j| \geq \lambda > \max_{j \in \bar{S}} |(A^\dagger b)_j|. \tag{2.27}$$

Proof. First, observe from the definitions of x^0 and S^0 that

$$S^0 = S \iff \{j \in [n] : |(A^\dagger b)_j| \geq \lambda\} = S \iff \min_{j \in S} |(A^\dagger b)_j| \geq \lambda > \max_{j \in \bar{S}} |(A^\dagger b)_j|.$$

Assume that x^* can be recovered via the scheme in one step, i.e., $x^1 = x^*$. By the definition of S^1, it follows that

$$S^1 := \{j \in [n] : |x_j^1| \geq \lambda\} = \{j \in [n] : |x_j^*| \geq \lambda\} = S.$$

By the stopping criterion (see Remark 2.2), we have $S^1 = S^0$. Thus, $S^0 = S$, which implies (2.27).

Assume that (2.27) holds, i.e., $S^0 = S$. The assumption that $Ax^* = b$ implies that

$$\|Ax^* - b\|_2 = \min_{x\in\mathbb{R}^n:\mathrm{supp}(x)\subseteq S} \|Ax - b\|_2$$

since $\mathrm{supp}(x^*) \in S$ and the norm is zero. Since A is injective, we have uniqueness and

$$x^* = \operatorname*{argmin}_{x\in\mathbb{R}^n:\mathrm{supp}(x)\subseteq S} \|Ax - b\|_2 = \operatorname*{argmin}_{x\in\mathbb{R}^n:\mathrm{supp}(x)\subseteq S^0} \|Ax - b\|_2 = x^1,$$

i.e., x^* can be recovered via the scheme in one step. ∎

In practice, the matrix A and vector b may contain (additive) noise and other artifacts. The following corollary shows that under certain conditions, the support set of x can be recovered by the SINDy algorithm in one step.

COROLLARY 2.9. *Let x be a vector with $\mathrm{supp}(x) = S$, let $A \in \mathbb{R}^{m\times n}$ with $m \geq n$, and let $b := Ax$. Let $B \in \mathbb{R}^{m\times n}$ and $\eta \in \mathbb{R}^m$ be the perturbations for A and b, respectively, and consider the perturbed problem*

$$(A + B)y = b + \eta. \tag{2.28}$$

If we set

$$r := \frac{\min_{i\in S} |x_i|}{\max_{i\in S} |x_i|} \quad \textit{and} \quad \gamma := \frac{\|\eta\|_\infty}{\|x\|_\infty}$$

and assume that A satisfies

$$\|U^{-1}\|_\infty < \frac{r}{2(\|B\|_\infty + \gamma)}, \tag{2.29}$$

where U is an upper triangular matrix satisfying $A^TA = U^TU$, then there is a λ such that the solution x^1 constructed by (2.1) for the perturbed problem has the correct support, i.e., $\mathrm{supp}(x^1) = S$.

Proof. Let $x^0 := (A + B)^\dagger(b + \eta)$, i.e., x^0 is the least-squares solution to (2.28). Then, by [28], we have

$$\|x^0 - x\|_\infty \leq \mathrm{cond}(U)\, \frac{\|B\|_\infty}{\|A\|_\infty}\, \|x\|_\infty + \mathrm{cond}(U) \frac{\|\eta\|_\infty}{\|A\|_\infty}.$$

Using (2.29), we have

$$\begin{aligned}\|x^0 - x\|_\infty &\leq \mathrm{cond}(U)\, \frac{\|B\|_\infty}{\|A\|_\infty}\, \|x\|_\infty + \mathrm{cond}(U) \frac{\|\eta\|_\infty}{\|A\|_\infty} \\ &= \|U^{-1}\|_\infty \left(\|B\|_\infty \|x\|_\infty + \|\eta\|_\infty\right) < \frac{\min_{i\in S} |x_i|}{2},\end{aligned}$$

where the condition number is with respect to ℓ_∞. Next, we want to show that the support set of x^1 is equal to S. Consider the indices outside of S, i.e., for each $j \in \bar{S}$ we have

$$|(x^0)_j| = |(x^0)_j - x_j| \leq \|x^0 - x\|_\infty < \frac{\min_{i\in S} |x_i|}{2}.$$

Algorithm 2.1 The SINDy algorithm [6] for $Ax = b$.

Input: $m \geq n$; $A \in \mathbb{R}^{m\times n}$ with $\operatorname{rank}(A) = n$; $b \in \mathbb{R}^m$.
1: Set $k = 0$; Initialize $x^0 = A^\dagger b$ and $S^{-1} = \emptyset$;
2: Set $S^k = \{j \in [n] : |x_j^k| \geq \lambda\}$; Choose $\lambda > 0$ such that $S^0 \neq \emptyset$;
3: **while** $S^k \neq S^{k-1}$ **do**
4: $\quad x^{k+1} = \operatorname{argmin}\|Ax - b\|_2$ such that $\operatorname{supp}(x) \subseteq S^k$;
5: $\quad S^{k+1} = \{j \in [n] : |x_j^{k+1}| \geq \lambda\}$;
6: $\quad k = k + 1$;
7: **end while**
8: **Output:** x^k.

Thus, if we set $\lambda = \min_{i\in S} |x_i|/2$, then $x^1|_{\bar{S}} = 0$. If $j \in S$, then

$$|(x^0)_j| \geq |x_j| - |(x^0)_j - x_j| \geq \min_{i\in S} |x_i| - \|x^0 - x\|_\infty > \frac{\min_{i\in S} |x_i|}{2}$$

and thus $\operatorname{supp}(x^1) = S$. ∎

We summarize all of the convergence results in the following theorem. The algorithm proposed in [6] is summarized in Algorithm 2.1.

THEOREM 2.10. *Assume that $m \geq n$. Let $A \in \mathbb{R}^{m\times n}$ with $\|A\|_2 = 1$, $b \in \mathbb{R}^m$, and $\lambda > 0$. Let x^k be the sequence generated by* (2.1). *Define the objective function F by* (2.4). *We have*

(i) *x^k converges to a fixed point of the iterative scheme defined by* (2.1) *in at most n steps;*
(ii) *a fixed point of the scheme is a local minimizer of F;*
(iii) *a global minimizer of F is a fixed point of the scheme;*
(iv) *x^k strictly decreases F unless the iterates are stationary.*

The preceding convergence analysis for Algorithm 2.1 can be readily adapted to a variety of SINDy-based algorithms. For example, in [20], the authors proposed the Sequential Threshold Ridge regression (STRidge) algorithm to find a sparse approximation of the solution of $Ax = b$. Instead of minimizing the function F, the STRidge algorithm minimizes the objective function

$$F_1(x) := \|Ax - b\|_2^2 + \gamma\|x\|_2^2 + \lambda^2\|x\|_0, \quad x \in \mathbb{R}^n, \tag{2.30}$$

by iterating

$$x^0 = A^\dagger b, \tag{2.31a}$$

$$S^k = \{j \in [n] : |x_j^k| \geq \lambda\}, \quad k \geq 0 \tag{2.31b}$$

$$x^{k+1} = \operatorname*{argmin}_{x\in\mathbb{R}^n : \operatorname{supp}(x)\subseteq S^k} \|Ax - b\|_2^2 + \gamma\|x\|_2^2. \tag{2.31c}$$

We assume that the parameter $\gamma > 0$ is fixed. By defining

$$\tilde{A} := \begin{pmatrix} A \\ \gamma I \end{pmatrix} \in \mathbb{R}^{(m+n)\times n}, \quad \tilde{b} := \begin{pmatrix} b \\ 0 \end{pmatrix} \in \mathbb{R}^{m+n}, \tag{2.32}$$

then $F_1(x) = \|\tilde{A}x - \tilde{b}\|_2^2 + \lambda^2\|x\|_0$ is equivalent to the objective function of Algorithm 2.1 with $\tilde{A}$ and $\tilde{b}$. We then obtain the following corollary.

COROLLARY 2.11. *Let $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and $\gamma, \lambda > 0$. Assume that $\|\tilde{A}\|_2 = 1$, where $\tilde{A}$ is defined by* (2.32). *Let x^k be the sequence generated by* (2.31). *Define the objective function F_1 by* (2.30). *We have*

(i) *the iterates x^k converge to a fixed point of the iterative scheme defined by* (2.31);
(ii) *a fixed point of the scheme is a local minimizer of F_1;*
(iii) *a global minimizer of F_1 is a fixed point of the scheme;*
(iv) *the iterates x^k strictly decrease F_1 unless the iterates are stationary.*

Note that we no longer require A to be injective, since concatenation with the identity matrix makes $\tilde{A}$ injective.

2.3. Examples and sharpness. We construct a few examples to highlight the effects of different choices of $\lambda > 0$. In particular, we show that the scheme obtains nontrivial sparse approximations, give an example where the minimizer is obtained in one step, and provide an example in which the maximum number of steps (i.e., $n-1$ steps) is required.[2] In all examples, A is injective.

Example 2.12. Consider a lower-triangular matrix $A \in \mathbb{R}^{5\times 5}$ given by

$$A := \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -0.1 & 0.9 & 0 & 0 & 0 \\ -0.1 & -0.1 & 0.8 & 0 & 0 \\ -0.1 & -0.1 & -0.1 & 0.7 & 0 \\ -0.1 & -0.1 & -0.1 & -0.1 & 0.6 \end{pmatrix}.$$

Let $x, b \in \mathbb{R}^5$ be such that

$$x := \big(10, 0.95, 0.9, 0.85, 0.8\big)^T,$$
$$b := Ax = \big(10, -0.145, -0.375, -0.59, -0.79\big)^T.$$

We want to obtain a 1-sparse approximation of the solution x from the system $Ax = b$. First, observe that:

$$\min_{j\in S} |(A^\dagger b)_j| = 10, \quad \max_{j\in \bar{S}} |(A^\dagger b)_j| = 0.95,$$

where $S = \{1\}$. Thus by Proposition 2.8, choosing $\lambda \in (0.95, 10]$ will yield immediate convergence:

$$x^0 = \big(10, 0.95, 0.9, 0.85, 0.8\big)^T, \qquad S^0 = \{1\}, \tag{2.33a}$$
$$x^1 = \big(9.7981, 0, 0, 0, 0\big)^T, \qquad S^1 = \{1\}. \tag{2.33b}$$

Indeed, we obtain a 1-sparse approximation of x in one step. Now consider a parameter outside of the optimal range, for example, $\lambda = 0.802$. Applying Algorithm 2.1 to the linear system yields

$$x^0 = \big(10, 0.95, 0.9, 0.85, 0.8\big)^T, \qquad S^0 = \{1,2,3,4\}, \tag{2.34a}$$
$$x^1 = \big(9.9366, 0.8725, 0.8031, 0.7255, 0\big)^T, \qquad S^1 = \{1,2,3\}, \tag{2.34b}$$
$$x^2 = \big(9.8869, 0.8117, 0.7271, 0, 0\big)^T, \qquad S^2 = \{1,2\}, \tag{2.34c}$$
$$x^3 = \big(9.8417, 0.7566, 0, 0, 0\big)^T, \qquad S^3 = \{1\}, \tag{2.34d}$$
$$x^4 = \big(9.7981, 0, 0, 0, 0\big)^T, \qquad S^4 = \{1\}. \tag{2.34e}$$

[2]The code is available at https://github.com/linanzhang/SINDyConvergenceExamples.

Therefore, a 1-sparse approximation of x is obtained in four steps, which is the maximum number of iterations Algorithm 2.1 needs in order to obtain a 1-sparse approximation (see Corollary 2.3).

The following examples shows that the iterative scheme, (2.1), obtains fixed-points which are not obtainable via direct thresholding. In fact, if we reorder the support sets S^k based on the magnitude of the corresponding components, we observe that the locations of the correct indices will evolve over time. The provides evidence that, in general, iterating the scheme is required.

Example 2.13. Consider the matrix $A \in \mathbb{R}^{10\times 10}$ given by

$$A = \begin{pmatrix} 4 & 5 & 1 & 6 & 8 & 4 & 6 & 6 & 2 & 7 \\ 6 & 5 & 7 & 5 & 3 & 3 & 2 & 5 & 9 & 2 \\ 1 & 5 & 1 & 7 & 4 & 8 & 1 & 3 & 9 & 7 \\ 10 & 2 & 9 & 5 & 5 & 10 & 0 & 8 & 1 & 2 \\ 9 & 9 & 3 & 9 & 6 & 4 & 3 & 7 & 1 & 4 \\ 10 & 1 & 7 & 8 & 7 & 4 & 10 & 3 & 3 & 6 \\ 2 & 4 & 4 & 5 & 6 & 9 & 1 & 9 & 1 & 9 \\ 2 & 5 & 1 & 3 & 6 & 3 & 10 & 7 & 2 & 1 \\ 1 & 1 & 1 & 3 & 10 & 4 & 4 & 4 & 5 & 1 \\ 6 & 5 & 1 & 4 & 2 & 5 & 1 & 5 & 1 & 8 \end{pmatrix}.$$

Let $x, \eta, b \in \mathbb{R}^{10}$ be such that

$$\begin{aligned} x &:= \big(1, 1, 1, 0, 0, 0, 0, 0, 0, 0\big)^T, \\ \eta &:= \big(0.23, 0.08, -0.01, -0.02, 0.04, -0.28, -0.32, 0.09, 0.30, 0.63\big)^T, \\ b &:= Ax + \eta = \big(10.23, 18.08, 6.99, 20.98, 21.04, 17.72, 9.68, 8.09, 3.30, 12.63\big)^T, \end{aligned}$$

where each element of η is drawn independently and identically distributed (i.i.d.) from the normal distribution $\mathcal{N}(0, 0.25)$. We want to recover x from the noisy data b using Algorithm 2.1. The support set to be recovered is $S := \{1, 2, 3\}$. Setting $\lambda = 0.7$ in Algorithm 2.1 yields

$$\begin{aligned} &x^0 = \big(\mathbf{0.88}, \mathbf{2.83}, \mathbf{2.04}, -1.60, 0.84, 0.63, 0.13, -1.82, -0.42, 0.26\big)^T, \\ &S^0 = \{2, 3, 8, 4, 1, 5\}, \end{aligned} \tag{2.35a}$$

$$\begin{aligned} &x^1 = \big(\mathbf{1.06}, \mathbf{1.08}, \mathbf{0.96}, -0.10, 0.04, 0, 0, -0.03, 0, 0\big)^T, \\ &S^1 = \{2, 1, 3\}, \end{aligned} \tag{2.35b}$$

$$\begin{aligned} &x^2 = \big(\mathbf{1.04}, \mathbf{1.01}, \mathbf{0.94}, 0, 0, 0, 0, 0, 0, 0\big)^T, \\ &S^2 = \{1, 2, 3\}, \end{aligned} \tag{2.35c}$$

where each S^k is reordered such that the jth element of S^k is the jth largest (in magnitude) element in x^k. Note that we have highlighted the desired components in bold.

Several important observations can be made from this example. First, there is no choice of λ so that the method converges in one step, since the value of x^0_1 is smaller (in magnitude) than two components on $\bar{S}$; however, the method still terminates at the correct support set. Second, setting $\lambda > 0.9$ will remove the first component immediately, yielding an incorrect solution. Last, the order of the indices in the

support set changes between steps, which shows that the solution x^k is not simply generated by peeling off the smallest elements of $A^\dagger b$. These observations lead one to conclude that the iterative scheme is more refined than just choosing the most important terms from $A^\dagger b$, i.e., the iterations shuffle the components and help to locate the correct components.

In the following example, we provide numerical support for Theorem 2.4.

Example 2.14. In Table 1, we list the values of the objective function F for the different experiments, where F is defined by (2.4). Recall that in Theorem 2.4, we have assumed that $\|A\|_2 = 1$. Thus to compute $F(x^k)$ for a given example, one may need to rescale $Ax = b$ by $\|A\|_2$. It can be seen from Table 1 that the value of $F(x^k)$ strictly decreases in k.

TABLE 1
The value of the objective function F in different experiments.

Inputs A and b	Parameter λ	Outputs x^k and S^k	$F(x^k)$
As defined in Example 2.12	8	As given in (2.33)	$F(x^0) = 320.0000$ $F(x^1) = 65.2119$
As defined in Example 2.12	0.802	As given in (2.34)	$F(x^0) = 3.2160$ $F(x^1) = 2.7727$ $F(x^2) = 2.3688$ $F(x^3) = 2.0490$ $F(x^4) = 1.8551$
As defined in Example 2.13	0.7	As given in (2.35)	$F(x^0) = 4.9000$ $F(x^1) = 2.9401$ $F(x^2) = 1.4702$

In the following example, we examine the relationship between the recovery rate, the condition number, and the size of perturbations to the data (see Corollary 2.9).

Example 2.15. Consider the matrix $A \in \mathbb{R}^{5\times 5}$ given by

$$A := \frac{1}{\sqrt{1+\epsilon^2}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ \epsilon & 0 & 0 & 0 & 0 \\ 0 & \epsilon & 0 & 0 & 0 \\ 0 & 0 & \epsilon & 0 & 0 \\ 0 & 0 & 0 & \epsilon & 0 \end{pmatrix},$$

whose columns have unit ℓ^2-norm. The condition number scales like ϵ^{-1}. Consider the 1-sparse vector $x := \big(1, 0, 0, 0, 0\big)^T$ and set $b := Ax + \eta$, where each element of η is drawn i.i.d. from the normal distribution $\mathcal{N}(0, \sigma^2)$.

By varying ϵ and σ^2, we examine the recovery rate for various condition numbers and perturbations. For each pair (ϵ, σ^2), we solved $A\tilde{x} = b$ using Algorithm 2.1 for 1,000 trials and computed the probability of the exact recovery rate, i.e., $\mathbb{P}(\text{supp}(\tilde{x}) = \text{supp}(x))$.

It can be observed from Table 2 that if the matrix A is well-conditioned and the additive noise is relatively small, it is very likely that the iterates x^k generated by the SINDy algorithm converge to a vector $\hat{x}$ such that $\text{supp}(\hat{x}) = \text{supp}(x)$. On the other hand, when the matrix A is ill-conditioned and the additive noise is relatively large, it is unlikely that the SINDy algorithm generates a reasonable approximation of x.

TABLE 2

Example 2.15: *The computed probability of exact recovery over* 1,000 *trials, using Algorithm* 2.1. *There is a direct relationship between the conditioning of the matrix and the amount of noise that the model can tolerate.*

	$\text{cond}(A) = 10$	$\text{cond}(A) = 10^2$	$\text{cond}(A) = 10^4$	$\text{cond}(A) = 10^5$
$\sigma^2 \sim 10^{-6}$	1	1	1	1
$\sigma^2 \sim 10^{-5}$	1	1	1	0.54
$\sigma^2 \sim 10^{-4}$	1	1	0.52	0.005
$\sigma^2 \sim 10^{-2}$	1	0.54	0	0

3. Application: Model identification of dynamical systems. Let u be an observed dynamic process governed by a first-order system,

$$\dot{u}(t) = f(u(t)),$$

where f is an unknown nonlinear equation. One application of the SINDy algorithm is for the recovery (or approximation) of f directly from data. In this section, we apply Algorithm 2.1 to this problem and show that relatively accurate solutions can be obtained when the observed data is perturbed by a moderate amount of noise.

Before detailing the numerical experiments, we first define two relevant quantities used in our error analysis. Let $x \in \mathbb{R}^n$ be the (noise-free) coefficient vector, and let $\eta \in \mathbb{R}^n$ be the mean-zero noise. The signal-to-noise ratio (SNR) is defined by

$$\text{SNR}(x, \eta) := 10 \log_{10} \left(\frac{\text{var}(x)}{\text{var}(\eta)} \right) = 10 \log_{10} \left(\frac{\|x - \text{mean}(x)\|_2^2}{\|\eta\|_2^2} \right).$$

Given $Ax = b$, let x_{true} be the correct sparse solution that solves the noise-free linear system, and let x be the approximation of x_{true} returned by Algorithm 2.1. The relative error E of x is defined by

$$E(x) := \frac{\|x - x_{\text{true}}\|_2}{\|x_{\text{true}}\|_2}.$$

3.1. The Lorenz system. Consider the Lorenz system

$$\begin{cases} \dot{u}_1 = 10(u_2 - u_1), \\ \dot{u}_2 = u_1(28 - u_3) - u_2, \\ \dot{u}_3 = u_1 u_2 - \frac{8}{3} u_3, \end{cases} \tag{3.1}$$

which produces chaotic solutions. To generate the synthetic data for this experiment, we set the initial data $u(0) = \big(-5, 10, 30\big)^T$ and evolve the system using the Runge–Kutta method of order 4 up to time-stamp $T = 10$ with time step $h = 0.025$. The simulated data is defined as $u(t)$. The noisy data $\tilde{u}(t)$ is obtained by adding Gaussian noise directly to $u(t)$,

$$\tilde{u} = u + \eta, \quad \eta \sim \mathcal{N}(0, \sigma^2).$$

Let $A = A(\tilde{u}(t))$ be the dictionary matrix consisting of (tensorized) polynomials in $\tilde{u}$ up to order p:

$$A = \begin{pmatrix} | & | & | & | & & | \\ 1 & P(\tilde{u}(t)) & P^2(\tilde{u}(t)) & P^3(\tilde{u}(t)) & \cdots & P^p(\tilde{u}(t)) \\ | & | & | & | & & | \end{pmatrix},$$

where

$$P(\tilde{u}(t)) := \begin{pmatrix} | & | & | \\ \tilde{u}_1(t) & \tilde{u}_2(t) & \tilde{u}_3(t) \\ | & | & | \end{pmatrix}, \tag{3.2a}$$

$$P^2(\tilde{u}(t)) := \begin{pmatrix} | & | & | & | & | & | \\ \tilde{u}_1(t)^2 & \tilde{u}_1(t)\tilde{u}_2(t) & \tilde{u}_1(t)\tilde{u}_3(t) & \tilde{u}_2(t)^2 & \tilde{u}_2(t)\tilde{u}_3(t) & \tilde{u}_3(t)^2 \\ | & | & | & | & | & | \end{pmatrix}, \tag{3.2b}$$

$$P^3(\tilde{u}(t)) := \begin{pmatrix} | & | & | & | & & | \\ \tilde{u}_1(t)^3 & \tilde{u}_1(t)^2\tilde{u}_2(t) & \tilde{u}_1(t)^2\tilde{u}_3(t) & \tilde{u}_1(t)\tilde{u}_2(t)^2 & \cdots & \tilde{u}_3(t)^3 \\ | & | & | & | & & | \end{pmatrix}, \tag{3.2c}$$

and so on. Each column of the matrices in (3.2) is a particular polynomial (candidate function) and each row is a fixed time-stamp. Let b be the numerical approximation of $\dot{u}$:

$$b_i(kh) := \begin{cases} \dfrac{\tilde{u}_i(h) - \tilde{u}_i(0)}{h} & \text{if } kh = 0, \\ \dfrac{\tilde{u}_i((k+1)h) - \tilde{u}_i((k-1)h)}{2h} & \text{if } 0 < kh < T, \\ \dfrac{\tilde{u}_i(T) - \tilde{u}_i(T-h)}{h} & \text{if } kh = T \end{cases} \tag{3.3}$$

for $i = 1, 2, 3$. Note that b is approximated directly from the noisy data, so it will be inaccurate (and likely unstable). We want to recover the governing equation for the Lorenz system (i.e., the right-hand side of (3.1)) by finding a sparse approximation to solution of the linear system $Ax = b$ using Algorithm 2.1.

With $p = 5$ and $\lambda = 0.8$, we apply Algorithm 2.1 on data with different noise levels. The resulting approximations for x (the coefficients) are listed in Table 3. The identified systems are

(i) $\sigma^2 = 0.1$ (where $\mathrm{SNR}(u, \eta) = 41.1508$):

$$\begin{cases} \dot{u}_1 = -9.8122\, u_1 + 9.8163\, u_2, \\ \dot{u}_2 = 27.1441\, u_1 - 0.8893\, u_2 - 0.9733\, u_1\, u_3, \\ \dot{u}_3 = -2.6238\, u_3 + 0.9841\, u_1\, u_2 \end{cases} \tag{3.4}$$

with $E(x) = 0.0278$;

(ii) $\sigma^2 = 0.5$ (where $\mathrm{SNR}(u, \eta) = 27.0682$):

$$\begin{cases} \dot{u}_1 = -9.7012\, u_1 + 9.6980\, u_2, \\ \dot{u}_2 = 27.0504\, u_1 - 0.8485\, u_2 - 0.9717\, u_1\, u_3, \\ \dot{u}_3 = -2.6197\, u_3 + 0.9834\, u_1\, u_2 \end{cases} \tag{3.5}$$

with $E(x) = 0.0334$.

To compare between the identified and true systems in the presence of additive noise on the observed data, we simulate the systems up to time-stamps $t = 20$ and $t = 100$. The resulting trajectories are shown in Figures 1 and 2.

Figure 1a shows that for a relatively small amount of noise ($\sigma^2 = 0.1$) the trajectory of the identified system almost coincides with the Lorenz attractor for a short time, specifically from $t = 0$ to about $t = 5$. On the other hand, Figure 1b shows

TABLE 3
Lorenz system: *The recovered coefficients for two noise levels.*

A	$\sigma^2 = 0.1$			$\sigma^2 = 0.5$		
	$\dot{u}_1$	$\dot{u}_2$	$\dot{u}_3$	$\dot{u}_1$	$\dot{u}_2$	$\dot{u}_3$
1	0	0	0	0	0	0
u_1	−9.8122	27.1441	0	−9.7012	27.0504	0
u_2	9.8163	−0.8893	0	9.6980	−0.8485	0
u_3	0	0	−2.6238	0	0	−2.6197
u_1^2	0	0	0	0	0	0
u_1u_2	0	0	0.9841	0	0	0.9834
u_1u_3	0	−0.9733	0	0	−0.9717	0
u_2^2	0	0	0	0	0	0
u_2u_3	0	0	0	0	0	0
u_3^2	0	0	0	0	0	0
u_1^3	0	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
u_3^5	0	0	0	0	0	0

that for a larger amount of noise ($\sigma^2 = 0.5$) the error between the trajectories of the identified system and the Lorenz attractor remains small for a shorter time (up to about $t = 4$). As expected, increasing the amount of noise will cause larger errors on the estimated parameters, and thus on the predicted trajectories. In both cases, the algorithm picks out the correct terms in the model. Increasing the noise will eventually lead to incorrect solutions.

3.2. The Thomas system. Consider the Thomas system:

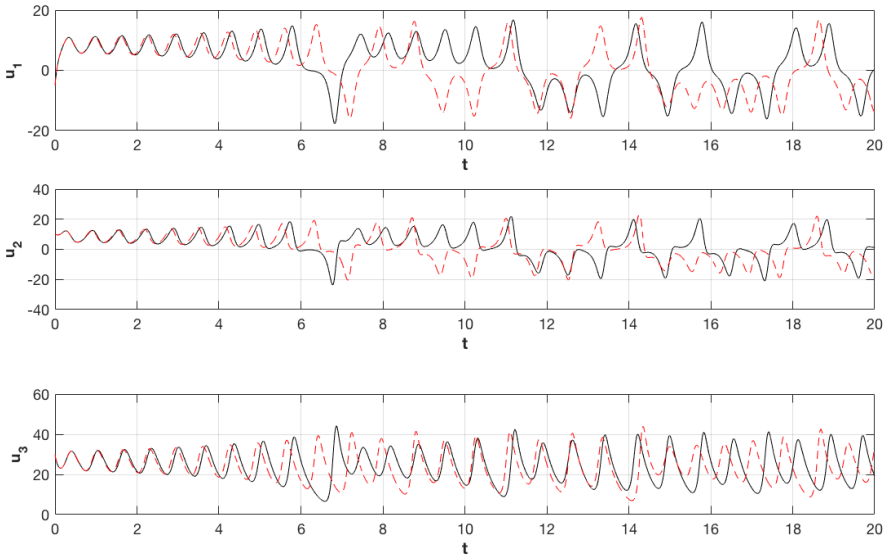
$$\begin{cases} \dot{u}_1 = -0.18u_1 + \sin(u_2), \\ \dot{u}_2 = -0.18u_2 + \sin(u_3), \\ \dot{u}_3 = -0.18u_3 + \sin(u_1), \end{cases} \tag{3.6}$$

which is a nonpolynomial system whose trajectories form a chaotic attractor. We simulate $u(t)$ using the initial condition $u(0) = \big(1, 1, 0\big)^T$ and by evolving the system using the Runge–Kutta method of order 4 up to time-stamp $t = 100$ with time step $h = 0.025$. We then add Gaussian noise to u and obtain the observed noisy data $\tilde{u}(t)$:

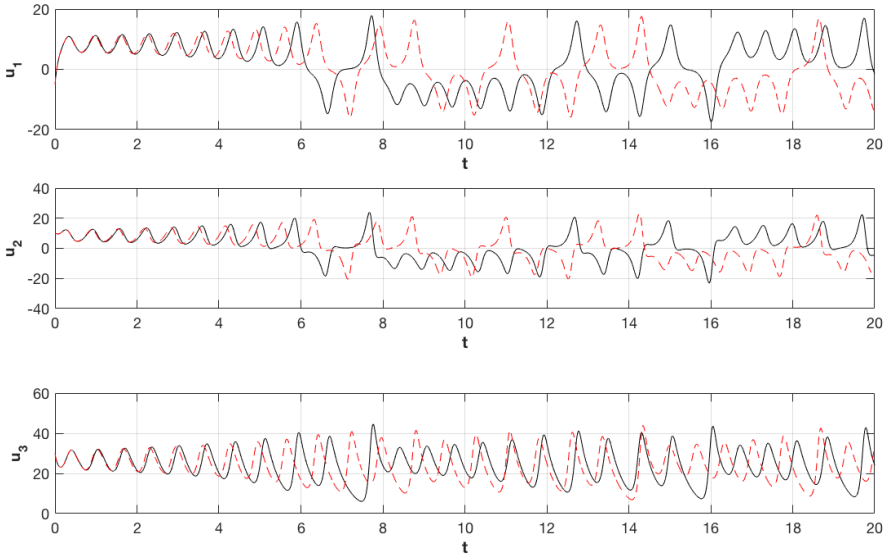
$$\tilde{u} = u + \eta, \quad \eta \sim \mathcal{N}(0, \sigma^2).$$

Let b be the numerical approximation of $\dot{u}$ which is defined by (3.3). To identify the governing equation for the data generated by the Thomas system (e.g., the right-hand side of (3.6)), we apply the algorithm to the linear system whose dictionary matrix $A = A(\tilde{u}(t))$ consists of three submatrices:

$$A = \begin{pmatrix} | & | & | \\ A_P & A_{\sin} & A_{\cos} \\ | & | & | \end{pmatrix}, \tag{3.7}$$



(a) $\sigma^2 = 0.1$



(b) $\sigma^2 = 0.5$

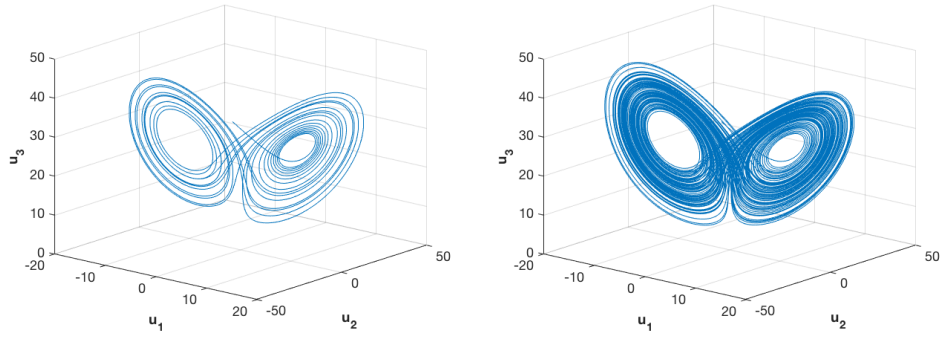
FIG. 1. Lorenz system*: Componentwise evolution of the trajectories. Solid line: the trajectory of the identified systems defined by* (a) (3.4) *and* (b) (3.5)*, respectively. Red dashed line: the "true" Lorenz attractor. (Figure in color online.)*

where

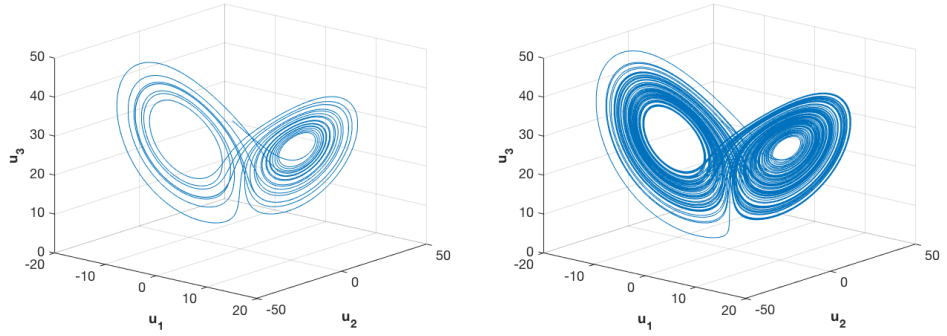
$$A_P = \begin{pmatrix} | & | & | & | & & | \\ 1 & P(\tilde{u}(t)) & P^2(\tilde{u}(t)) & P^3(\tilde{u}(t)) & \cdots & P^{p_1}(\tilde{u}(t)) \\ | & | & | & | & & | \end{pmatrix},$$

$$A_{\sin} = \begin{pmatrix} | & | & | & & | \\ \sin\left(P(\tilde{u}(t))\right) & \sin\left(P^2(\tilde{u}(t))\right) & \sin\left(P^3(\tilde{u}(t))\right) & \cdots & \sin\left(P^{p_2}(\tilde{u}(t))\right) \\ | & | & | & & | \end{pmatrix},$$

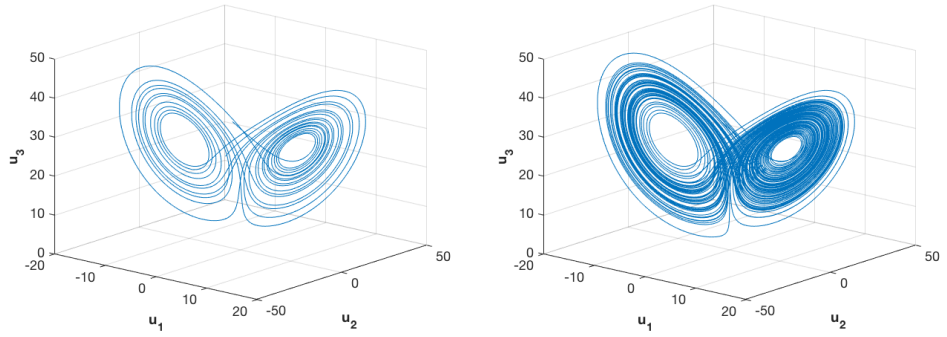
$$A_{\cos} = \begin{pmatrix} | & | & | & & | \\ \cos\left(P(\tilde{u}(t))\right) & \cos\left(P^2(\tilde{u}(t))\right) & \cos\left(P^3(\tilde{u}(t))\right) & \cdots & \cos\left(P^{p_3}(\tilde{u}(t))\right) \\ | & | & | & & | \end{pmatrix}.$$



(a) Lorenz attractor



(b) The trajectory defined by (3.4)



(c) The trajectory defined by (3.5)

FIG. 2. Lorenz system: *Trajectories of the Lorenz system from $t = 0$ to $t = 20$ (left column) and from $t = 0$ to $t = 100$ (right column).* (a) *The "true" Lorenz attractor defined by* (3.1). (b) *The trajectory defined by* (3.4)*, which is identified from data with additive noise* $\sigma^2 = 0.1$. (c) *The trajectory defined by* (3.5)*, which is identified from data with additive noise* $\sigma^2 = 0.5$.

Here, P^p is defined by (3.2), which denotes the matrix consisting of polynomials in $\tilde{u}$ of order p. The matrices $\sin(P^p)$ and $\cos(P^p)$ are obtained by applying the sine and cosine functions to each element of P^p, respectively.

With $p_1 = 3$, $p_2 = p_3 = 1$, and $\lambda = 0.1$, we apply the algorithm to data with different noise levels. The resulting approximations for x are listed in Table 4. The

TABLE 4
Thomas system: *The recovered coefficients for two noise levels.*

A	$\sigma^2 = 0.1$			$\sigma^2 = 0.5$		
	$\dot{u}_1$	$\dot{u}_2$	$\dot{u}_3$	$\dot{u}_1$	$\dot{u}_2$	$\dot{u}_3$
1	0	0	0	0	0	0
u_1	−0.1805	0	0	−0.1835	0	0
u_2	0	−0.1799	0	0	−0.1848	0
u_3	0	0	−0.1803	0	0	−0.1725
u_1^2	0	0	0	0	0	0
u_1u_2	0	0	0	0	0	0
u_1u_3	0	0	0	0	0	0
u_2^2	0	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
u_3^3	0	0	0	0	0	0
$\sin(u_1)$	0	0	0.9992	0	0	0.9658
$\sin(u_2)$	1.0014	0	0	1.0304	0	0
$\sin(u_3)$	0	1.0038	0	0	0.9956	0
$\cos(u_1)$	0	0	0	0	0	0
$\cos(u_2)$	0	0	0	0	0	0
$\cos(u_3)$	0	0	0	0	0	0

identified systems are

(i) $\sigma^2 = 0.1$ (where $\mathrm{SNR}(u, \eta) = 25.8469$):

$$\begin{cases} \dot{u}_1 = -0.1805u_1 + 1.0014\sin(u_2), \\ \dot{u}_2 = -0.1799u_2 + 1.0038\sin(u_3), \\ \dot{u}_3 = -0.1803u_3 + 0.9992\sin(u_1) \end{cases} \tag{3.8}$$

with $E(x) = 0.0023$;

(ii) $\sigma^2 = 0.5$ (where $\mathrm{SNR}(u, \eta) = 11.8738$):

$$\begin{cases} \dot{u}_1 = -0.1835u_1 + 1.0304\sin(u_2), \\ \dot{u}_2 = -0.1848u_2 + 0.9956\sin(u_3), \\ \dot{u}_3 = -0.1725u_3 + 0.9658\sin(u_1) \end{cases} \tag{3.9}$$

with $E(x) = 0.0267$.

Observe that the identified system defined by (3.8) is exact up to two significant digits. We simulate this system up to time-stamps $t = 200$ and $t = 1000$ and compare it with the trajectories of the Thomas system. We show the short-time evolution of the trajectories in Figure 3 and the long-time dynamics in Figure 4. It can be observed that although the coefficients are not exact, the trajectory of the identified system traces out a region similar to the exact trajectory in state-space.

Suppose that the dictionary matrix A defined in (3.7) does not contain the trigonometric terms present in the Thomas system, i.e., $p_2 = p_3 = 0$. In this example, we

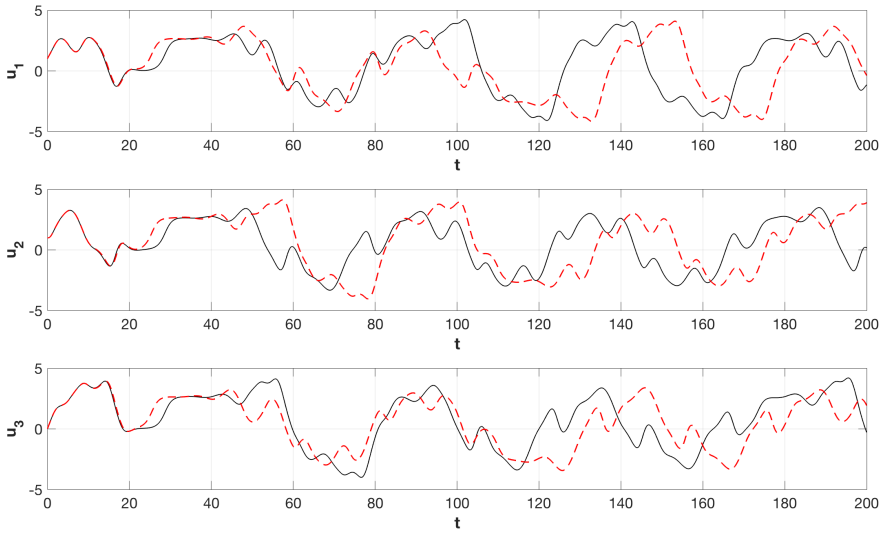
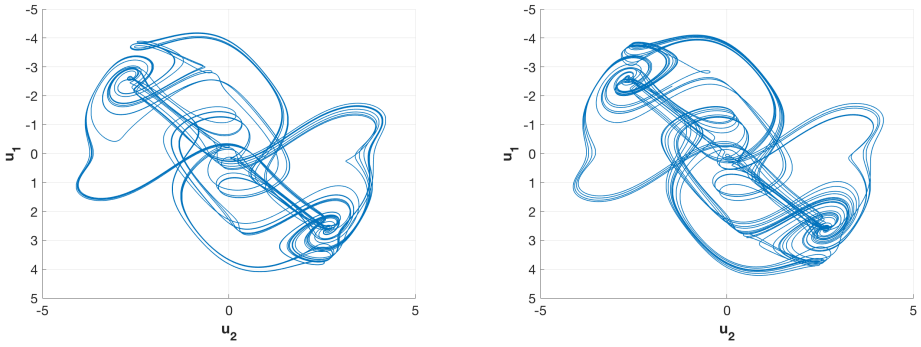


FIG. 3. Thomas system*: Componentwise evolution of the trajectories. Solid line: the trajectory of the identified system defined by* (3.8)*. Red dashed line: the "true" Thomas trajectory. (Figure in color online.)*



(a) The Thomas trajectory (b) The trajectory defined by (3.8)

FIG. 4. Thomas system: *Trajectories of the learned and "true" Thomas system from* $t = 0$ *to* $t = 1000$ *(right column).* (a) *The Thomas trajectory defined by* (3.6). (b) *The trajectory defined by* (3.8)*, which is identified from data with additive noise* $\sigma^2 = 0.1$.

consider a polynomial dictionary of order 5 which should yield the following equation:

$$\begin{cases} \dot{u}_1 = -0.18u_1 + u_2 - \dfrac{u_2^3}{6} + \dfrac{u_2^5}{120}, \\ \dot{u}_2 = -0.18u_2 + u_3 - \dfrac{u_3^3}{6} + \dfrac{u_3^5}{120}, \\ \dot{u}_3 = -0.18u_3 + u_1 - \dfrac{u_1^3}{6} + \dfrac{u_1^5}{120}. \end{cases} \tag{3.10}$$

The results for recovering the coefficients of (3.10) when the data is generated by (3.6)

are shown in Table 5. Since $1/120 \approx 0.0083$, λ must balance between the noise and the coefficients. When the noise level is low ($\sigma^2 = 0.01$), the algorithm is able to obtain a good approximation of the true coefficient matrix associated with (3.10) using a relatively small λ. This is illustrated by the first example in Table 5. When the noise level increases ($\sigma^2 = 0.05$), we observed that if λ is small enough such that the SINDy algorithm produces a solution that contains elements from the correct support set, it also picks up the noise. On the other hand, if λ is large enough to avoid the noise, then the SINDy algorithm is only able to pick up a subset of the correct terms. This is illustrated by the second example in Table 5.

TABLE 5

Approximating the Thomas system with polynomials; see (3.10)*: The recovered coefficients with two different sets of parameters* (σ^2, λ)*. The coefficients that appear are related to the Taylor approximation.*

A	$\sigma^2 = 0.01$, $\lambda = 0.003$			$\sigma^2 = 0.05$, $\lambda = 0.05$		
	$\dot{u}_1$	$\dot{u}_2$	$\dot{u}_3$	$\dot{u}_1$	$\dot{u}_2$	$\dot{u}_3$
1	0	0	0	0	0	0
u_1	−0.1839	0	0.9722	−0.2279	0	0.7778
u_2	0.9448	−0.1815	0	0.6844	−0.1981	0
u_3	0	0.9651	−0.1810	0	0.7332	−0.1994
u_1^3	0	0	−0.1474	0	0	−0.0771
u_2^3	−0.1374	0	0	−0.0603	0	0
u_3^3	0	−0.1439	0	0	−0.0693	0
u_1^5	0	0	0.0049	0	0	0
u_2^5	0.0042	0	0	0	0	0
u_3^5	0	0.0046	0	0	0	0
Other terms	0	0	0	0	0	0
SNR(u, η)	45.9575			32.0153		
$E(x)$	0.0464			0.2816		

4. Discussion. The SINDy algorithm proposed in [6] has been applied to various problems involving sparse model identification from complex dynamics. In this work, we provided several theoretical results that characterized the solutions produced by the algorithm and provided the rate of convergence of the algorithm. The results included showing that the algorithm approximates local minimizers of the ℓ^0-penalized least-squares problem, and thus can be characterized through various sparse optimization results. In particular, the algorithm produces a minimizing sequence, which converges to a fixed-point rapidly, thereby providing theoretical support for the observed behavior. Several examples showed that the convergence rates are sharp. In addition, we showed that iterating the steps is required; in particular, it is possible to obtain solutions through iterating that cannot be obtained via thresholding of the least-squares solution.

In future work, we would like to better characterize the effects of noise, detailed in section 3. It would be useful to have a quantifiable relationship between the thresholding parameter, the noise, and the expected recovery error. There are strategies for tuning λ in the SINDy algorithm using information criteria to distinguish between learned models; see [14, 15, 13]. Since our results show that the parameter λ in the

SINDy algorithm is related to the weight in the ℓ^0-minimization, one may be able to use results from compressive sensing to tune λ; see [17, 32].

Acknowledgement. The authors would like to thank J. Nathan Kutz for helpful discussions.

REFERENCES

[1] T. Blumensath and M. E. Davies, *Iterative thresholding for sparse approximations*, J. Fourier Anal. Appl., 14 (2008), pp. 629–654.
[2] T. Blumensath and M. E. Davies, *Iterative hard thresholding for compressed sensing*, Appl. Comput. Harmon. Anal., 27 (2009), pp. 265–274.
[3] T. Blumensath, M. Yaghoobi, and M. E. Davies, *Iterative hard thresholding and ℓ^0 regularisation*, Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing, 3 (2007), pp. III–877.
[4] J. Bongard and H. Lipson, *Automated reverse engineering of nonlinear dynamical systems*, Proc. Nat. Acad. Sci. USA, 104 (2007), pp. 9943–9948.
[5] L. Boninsegna, F. Nånuske, and C. Clementi, *Sparse learning of stochastic dynamic equations*, J. Chem. Phys., 148 (2018), 241723.
[6] S. L. Brunton, J. L. Proctor, and J. N. Kutz, *Discovering governing equations from data by sparse identification of nonlinear dynamical systems*, Proc. Nat. Acad. Sci. USA, 113 (2016), pp. 3932–3937.
[7] S. L. Brunton, J. L. Proctor, and J. N. Kutz, *Sparse identification of nonlinear dynamics with control (SINDYc)*, IFAC-PapersOnLine, 49 (2016), pp. 710–715.
[8] M. Dam, M. Brøns, J. J. Rasmussen, V. Naulin, and J. S. Hesthaven, *Sparse identification of a predator-prey system from simulation data of a convection model*, Physics of Plasmas, 24 (2017), 022310.
[9] L. S. T. Ho, H. Schaeffer, G. Tran, and R. Ward, *Recovery Guarantees for Polynomial Approximation from Dependent Data with Outliers*, preprint https://arxiv.org/abs/1811.10115, 2018.
[10] E. Kaiser, J. N. Kutz, and S. L. Brunton, *Sparse identification of nonlinear dynamics for model predictive control in the low-data limit*, Proc. A, 474 (2018), 20180335.
[11] J.-C. Loiseau and S. L. Brunton, *Constrained sparse Galerki regression*, J. Fluid Mech., 838 (2018), pp. 42–67.
[12] Z. Long, Y. Lu, X. Ma, and B. Dong, *PDE-Net: Learning PDEs from data*, in Proceedings of the 35th International Conference on Machine Learning, ICML 2018, International Machine Learning Society (IMLS), 2018, pp. 5067–5078.
[13] N. M. Mangan, T. Askham, S. L. Brunton, J. N. Kutz, and J. L. Proctor, *Model selection for hybrid dynamical systems via sparse regression*, Proc. R. Soc. A., 475 (2019), 20180534, https://doi.org/10.1098/rspa.2018.0534.
[14] N. M. Mangan, S. L. Brunton, J. L. Proctor, and J. N. Kutz, *Inferring biological networks by sparse identification of nonlinear dynamics*, IEEE Trans. Mol. Biol. Multi-Scale Commun., 2 (2016), pp. 52–63.
[15] N. M. Mangan, J. N. Kutz, S. L. Brunton, and J. L. Proctor, *Model selection for dynamical systems via sparse regression and information criteria*, Proce. A, 473 (2017), 20170009.
[16] Y. Pantazis and I. Tsamardinos, *A unified approach for sparse dynamical system inference from temporal measurements*, Bioinformatics, btz065, in press, https://doi.org/10.1093/bioinformatics/btz065.
[17] R. G. B. Petros Boufounos, Marco F. Duarte, *Sparse signal reconstruction from noisy compressive measurements using cross validation*, in Proceedings of the IEEE/SP 14th Workshop on Statistical Signal Processing, 2007 (SSP'07), IEEE, 2007, pp. 299–303.
[18] M. Quade, M. Abel, J. N. Kutz, and S. L. Brunton, *Sparse identification of nonlinear dynamics for rapid model recovery*, Chaos, 28 (2018), 063116.
[19] M. Raissi and G. E. Karniadakis, *Hidden physics models: Machine learning of nonlinear partial differential equations*, J. Comput. Phys., 357 (2018), pp. 125–141.
[20] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, *Data-driven discovery of partial differential equations*, Sci. Adv., 3 (2017), e1602614.
[21] H. Schaeffer, *Learning partial differential equations via data discovery and sparse optimization*, Proc. A, 473 (2017), 20160446.
[22] H. Schaeffer and S. G. McCalla, *Sparse model selection via integral terms*, Phys. Rev. E, 96 (2017), 023302.

[23] H. Schaeffer, G. Tran, and R. Ward, *Extracting sparse high-dimensional dynamics from limited data*, SIAM J. Appl. Math., 78 (2019), pp. 3279–3295, https://doi.org/10.1137/18M116798X.
[24] H. Schaeffer, G. Tran, and R. Ward, *Learning Dynamical Systems and Bifurcation Via Group Sparsity*, preprint, https://arxiv.org/abs/1709.01558, 2017.
[25] H. Schaeffer, G. Tran, R. Ward, and L. Zhang, *Extracting Structured Dynamical Systems Using Sparse Optimization with Very Few Samples*, preprint, https://arxiv.org/abs/1805.04158, 2018.
[26] M. Schmidt and H. Lipson, *Distilling free-form natural laws from experimental data*, Science, 324 (2009), pp. 81–85.
[27] M. Sorokina, S. Sygletos, and S. Turitsyn, *Sparse identification for nonlinear optical communication systems: SINO method*, Opt. Express, 24 (2016), 30443.
[28] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Texts Appl. Math. 12, Springer-Verlag, New York, 2013.
[29] R. Tibshirani, *Regression shrinkage and selection via the lasso*, J. Roy. Statist. Soc. Ser. B, 58 (1996), pp. 267–288.
[30] G. Tran and R. Ward, *Exact recovery of chaotic systems from highly corrupted data*, Multiscale Model. Simul., 15 (2017), pp. 1108–1129, https://doi.org/10.1137/16M1086637.
[31] J. A. Tropp, *Just relax: Convex programming methods for identifying sparse signals in noise*, IEEE Trans. Inform. Theory, 52 (2006), pp. 1030–1051.
[32] R. Ward, *Compressed sensing with cross validation*, IEEE Trans. Inform. Theory, 55 (2009), pp. 5773–5782.