

Globalizing the Carleman linear embedding method for nonlinear dynamics

I. Novikau¹ and I. Joseph¹

Lawrence Livermore National Laboratory, Livermore, California 94550, USA

(*Electronic mail: joseph5@llnl.gov)

(Dated: 20 October 2025)

The Carleman embedding method is a widely used technique for linearizing a system of nonlinear differential equations, but fails to converge in regions where there are multiple fixed points. We propose and test three different versions of a global piecewise Carleman embedding technique, based on partitioning space into multiple regions where the center and size of the embedding region are chosen to control convergence. The first method switches between local linearization regions of fixed size once the trajectory reaches the boundary of the current linearization chart. During the transition, the embedding is reconstructed within the newly created chart, centered at the transition point. The second method also adapts the chart size dynamically, enhancing accuracy in regions where multiple fixed points are located. The third method partitions the state space using a static grid with precomputed linearization charts of fixed size, making it more suitable for applications that require high speed. All techniques are numerically tested on multiple integrable and chaotic nonlinear dynamical systems demonstrating their applicability for problems that are completely intractable for the standard Carleman embedding method. Simulations of chaotic dynamical systems such as various types of strange attractors demonstrate the power of the adaptive methods, if a sufficiently low tolerance is imposed. Still, the non-adaptive version of the method, with fixed centers and sizes of the linearization charts, can be faster in simulating dynamical systems while providing similar accuracy and may be more appropriate as the basis of algorithms for future quantum computers.

I. INTRODUCTION

A. Motivation

Carleman linearization¹ is a widely used technique for embedding nonlinear dynamics within an infinite dimensional linear system of differential equations^{2,3}. The embedding of nonlinear ordinary differential equations (ODE) into an infinite-dimensional linear system is performed by augmenting the simulated state with monomials of the coordinates with ever increasing polynomial degree. Extensions of the standard Carleman technique use other libraries of basis functions such as Fourier series or orthogonal polynomials, like the Chebyshev polynomials. If the Carleman method converges, then one can substitute a nonlinear system for a much larger linear system, which can lead to practical advantages for solution speed and accuracy.

However, a serious drawback is that the convergence of the Carleman method has severe limitations that limit its ability to handle truly nonlinear dynamics as well as vast domains of initial conditions⁴⁻⁶. For example, the Carleman method must normalize the equations so that the solution remains in the interior of the unit disk in the complex plane and will fail if the solution tries to cross the boundary of the unit disk. More disconcerting is that the Carleman method will fail whenever the system has more than one fixed point inside the unit disk and the solution is too far from the linearization point at the center of the disk. A simple proof of this fact is presented for 1D flows in Appendix A. Since a linear system can only ever have a single fixed point, it should not be surprising that the Carleman linearizing transformation typically fails when the nonlinear system has multiple fixed points. Yet, this clearly limits the domain of applicability of the Carleman method to *weakly nonlinear dynamics*

and to initial conditions that are *attracted to a region within the unit disk*.

The goal of this work is to *globalize* the Carleman method so that it can be applied to more general classes of nonlinear dynamical systems and much wider domains for the initial conditions. Our ultimate motivation is to enable the use of a globalized Carleman method for advanced applications like learning the Koopman mode decomposition (KMD)⁷⁻⁹ for prediction and control^{10,11}, and for emerging computational paradigms that are optimized for linear systems such as tensor networks and quantum algorithms⁶.

Newton's method is another example of a nonlinear solution method that uses the assumption of linearizability as an essential part of the solution strategy. Yet, Newton's method fails when the initial guess is too far from the fixed point because the linear approximation is no longer accurate. Newton's method can be rescued with a globalization method like line search, dog leg search, or trust region search that ensures that the proposed step is limited to a size that actually improves the residual.

The finite element and spectral element methods were developed in order to handle similar globalization issues. Generic series expansions only have a finite radius of convergence that depends on the location of the basepoint of the expansion. Thus, in order to represent the solution in a manner that converges everywhere, it is necessary to cover the region of interest with multiple local domains (called elements) that have different series expansions within each local domain.

Here, we develop a globalized linearization method with a similar strategy of using multiple linearization regions to ensure that the trajectory is accurately represented over the entire time interval of interest. Each valid linearization region is called a *linearization chart*. In many cases of practical interest, the entire solution manifold can be covered with

a countable number of linearization charts, which form a *linearization atlas* for the solution manifold. Each linearizing transformation is assumed to be invertible within each chart, which allows one to uniquely transform between charts. Unlike the case of a piecewise linear manifold, here, the transition functions between linearization charts are allowed to either be smooth nonlinear functions of class $\mathcal{C}^k$ or piecewise continuous functions of class $\mathcal{C}^0$. This allows one to define the linearization atlas as either being smooth of class $\mathcal{C}^k$ or piecewise continuous of class $\mathcal{C}^0$, respectively.

Depending on how much is known about the system of ODEs, the linearization charts can either be precomputed in advance or, if the vector field is assumed to be given by a black-box oracle, computed on the fly. Clearly, for the Carleman method, a new chart must be used each time the solution leaves the Carleman unit disk. Moreover, the radius of the unit disk may need to be rescaled as the next chart is accessed in order to optimize accuracy and ensure convergence. Thus, a well defined algorithmic approach is needed to guarantee that a target error budget can be achieved.

This work explores these ideas and issues with a proposed series of algorithms tested on realistic examples of integrable and chaotic dynamical systems. We numerically demonstrate that the globalized Carleman method can be extended to a wide range of nonlinear dynamical systems that cannot be treated with the standard approach. Examples where the globalized version succeeds but the standard approach fails include 1D systems with multiple fixed points, stable limit cycles in the 2D plane, and 3D systems with strange attractors, including the Rössler, Lorentz, and Chen's attractors. While this demonstrates the great power of the globalized Carleman method, our understanding of how to develop optimal Carleman atlases has only just begun.

One of the key points that we are making is that the linearization of a flow on a manifold does not require a complete KMD. Rather, *the flow can be linearized using any appropriate set of basis functions if and only if this set of basis functions can capture the correct topological structure needed to represent the flow on the manifold*. While a complete generalized KMD, including singular Dirac-delta-like eigendistributions, clearly provides a complete and simple representation for the Koopman evolution operator, it is not analytic and in fact becomes singular at the boundaries of independent dynamically invariant sets. Thus, if the goal is to discover the KMD through numerical computation, then it is important to first linearize the flow with a numerically well-behaved basis set from which one can then determine the KMD.

B. The Carleman and Koopman embedding methods

The Carleman method was derived just after the work of Koopman¹² and von Neumann¹³ on the linearization of Hamiltonian dynamics. The general linearizing transformation for a set of ordinary differential equations (ODEs) can always be understood as a linear partial differential equation (PDE) for the conservation of the probability distribution function (PDF) of the set of solutions^{6,14}. This linear

conservation law can always be expressed as an infinite dimensional linear matrix equation, by computing its components over a set of basis functions that is complete over phase space. In fact, the Carleman method has been proven to simply be a complex analytic version of the Koopman-von Neumann (KvN) approach (see 6 Appendix B).

The Carleman method has recently risen to great prominence in the quantum algorithms community. Quantum algorithms were developed that efficiently implement the Carleman embedding technique^{15–17} and enable the efficient simulation of *weakly nonlinear dissipative* dynamical systems on quantum computers. An important advancement was the ability to rigorously prove the convergence of the quantum Carleman algorithm for the first time¹⁵. However, convergence can only be proven (1) if the nonlinearity is sufficiently weak and (2) given the promise that the initial condition is sufficiently close to an attracting fixed point (e.g. see 15). Assuming the linearly embedded system converges to a fixed point, it can be solved efficiently on a quantum computer using quantum Hamiltonian simulation methods such as quantum linear system solvers^{18–21} or the more recent Linear Combination of Hamiltonian Simulation^{22–24} method.

In fact, the first proposed quantum embedding technique for nonlinear dynamics was the KvN method¹⁴ and it is not restricted to dissipative or weakly nonlinear dynamics. The KvN method has been successfully employed to represent nonlinear dynamics through augmented linear systems in several specific applications^{5,14,25,26}. Explicit quantum algorithms with rigorous convergence guarantees were recently developed for the first time in 26 and 27. Today, both embedding techniques are widely discussed as promising candidates for designing quantum algorithms for fluid dynamics^{28–32}, kinetic theory^{6,17,33,34}, and, with certain caveats, general nonlinear differential equations^{5,15,16,26,35,36}.

A few of the main limitations of the Carleman embedding technique are that it is inherently local, has a limited radius of convergence, and only operates correctly in the vicinity of a single fixed point. Importantly, it cannot globally represent a system with multiple distinct fixed points. If the problem is initialized near two or more fixed points, or if the trajectory leaves the initial linearization region, the Carleman method fails to accurately simulate the system's evolution and may become unstable.

Several techniques were recently proposed to provide global embedding capturing dynamics near multiple equilibrium points. One option is to use a basis other than monomials for linearization. For instance, one can use Fourier basis functions to enable the embedding to capture periodic or oscillatory dynamics³⁷. In the so-called super-linearization approach^{38,39}, additional variables beyond simple monomials are introduced using, in general, non-polynomial coordinate transformations. The disadvantage of this approach is that it is still not guaranteed to converge, and, in fact, the linear system becomes more poorly conditioned as the library of functions increases in expressivity. Moreover, the conditions under which a general dynamical system can be super-linearized remain an open question. Apart from this, machine learning techniques also can be used for constructing global linearization of nonlinear systems⁴⁰, again, with

some improvement in performance but without guarantees of convergence.

The key idea behind the global Carleman embedding technique, which is more suitable for truly nonlinear applications, is piecewise linearization, where the method switches to a new local linearization chart once the convergence radius is exceeded. While this idea was mentioned as the motivation for 41, the authors did not describe an algorithm for switching between charts, and, hence, an explicit switching procedure was not proposed, implemented, or tested.

The authors of 42 and 43 demonstrated that the precision of Carleman embedding can be improved by linearizing the nonlinear system around its initial conditions. They proved that, in general, this doubles the order of the polynomial for which the Carleman method can exactly represent the solution. If one partitions the space into separate linearization regions, which we call charts, and applies a local Carleman embedding within each chart, the overall precision of the method can be improved. However, such piecewise embeddings require careful detection of the moment when it is necessary to switch between charts along the evolution of the trajectory.

C. Connection to Koopman theory

Carleman linearization can be regarded as a special case of Koopman theory^{8,12}. In the latter, one seeks a linear operator, namely, the Koopman operator, that acts on a set of observables such that the underlying nonlinear dynamics become linear on the span of the chosen observables. The Carleman technique is simply an explicit construction of the Koopman operator using monomials of the state as observables^{44,45}. Koopman theory is more general than Carleman embedding and can potentially extend the regions where the linear representation accurately emulates the nonlinear dynamics. However, for an arbitrary nonlinear system, Koopman theory does not guarantee that a smooth global transformation exists that converges everywhere because any dynamically invariant linearizing transformation must become singular near fixed points and strange attractors⁴⁶. For example, cases with multiple isolated equilibria are particularly challenging for many embedding methods^{47,48}. However, even in such cases, counterexamples exist where a smooth global embedding can be constructed³⁸.

In the KMD⁷, each observable is represented as an expansion in a basis of the Koopman operator's eigenfunctions, where the coefficients in the expansion are called Koopman modes. Numerical algorithms such as Dynamic Mode Decomposition (DMD)⁴⁹ approximate KMD in real-world applications using experimental or numerically computed data. The finite-dimensional linear model produced by DMD can be interpreted as a projection of the true Koopman operator onto the span of the observed snapshots. While Carleman requires explicit nonlinear equations to construct the linear embedding, the DMD technique relies on measured data, such as time-series data, to approximate observed nonlinear dynamics with a finite linear model. Ultimately, both Carleman linearization and DMD/KMD methodologies yield lin-

ear representations that serve as approximations for the simulation and analysis of inherently nonlinear dynamical systems.

Koopman theory and DMD techniques are widely employed in the analysis of complex nonlinear problems^{50–52}, especially for fluid dynamics^{8,53}. In fact, the well known Sparse Identification of Nonlinear Dynamics (SINDy) technique⁵¹ is a data-driven method for discovering a sparse library of functions that are suitable for describing the dynamics of interest.

For example, in the field of plasma physics, these techniques have been applied to the analysis of both numerical simulations and experiments^{54–58}. In Ref. 59, the authors utilized DMD to characterize the spatial and temporal behavior of plasma instabilities in a spheromak plasma device. In Refs. 60 and 61, the authors investigated DMD-based dimensionality reduction for high-dimensional plasma simulations. In Refs. 62 and 63, DMD was employed to accelerate plasma simulations. This technique was further applied in Ref. 64 to construct a reduced numerical magnetohydrodynamics (MHD) model. In Ref. 65, DMD was used to extract dominant drift-wave modes from gyrokinetic simulations and was benchmarked against results obtained from gyrokinetic eigensolver simulations.

Linear embeddings have also found widespread use in machine learning^{53,66} and robotics⁶⁷, where high-dimensional and nonlinear dynamics are common. A good example is the forecasting of the time evolution of nonlinear problems using Koopman-based autoencoder networks^{68,69}. Another example is the use of the Koopman operator in reinforcement learning to estimate the value function, thereby effectively eliminating the dependence on future system states⁷⁰.

One of the key advantages of applying the Koopman operator in machine learning is better interpretability of input-output relationships, in contrast to neural networks, which are often regarded as black-box models^{71,72}. Linear embeddings are also beneficial in the context of optimal control problems^{11,73–76}. In particular, linear Koopman Model Predictive Control (MPC)¹⁰ leads to convex optimization problems with a unique global solution. In this context, the Koopman operator can be approximated using deep neural networks, enabling its subsequent application in the optimal control of dynamical systems⁷⁷. However, one should be careful when attempting to learn linear embeddings blindly, as there are fundamental limitations which constrain the applicability of linearization techniques^{47,48,78,79}.

D. What is the Optimal Linearization Atlas?

The number of linearization charts needed to cover the manifold with a linearization atlas are determined by at least two separate considerations: *topology* and *analyticity*. Koopman charts are only constrained by topology, because they can use any library of functions to define the linearization map. In contrast, Carleman charts make a specific choice for the basis functions, namely polynomials in a specific coordinate system that are centered at a particular basepoint of the manifold, and, hence, the Carleman embedding proce-

ture will only converge if the linearization map converges when written in terms of this basis. In DMD terminology, one would refer to a convergent Koopman linearization process as an extended- or super-Carleman embedding. However, a better description is simply that the Carleman embedding is a specific type of Koopman embedding.

When using Koopman theory, one should also distinguish between two separate notions of linearizability. The first notion of linearizability is given by the *flow-box rectification* theorem, which states that a vector field can always be rectified in any region where it has no singular points, i.e. fixed-points where it vanishes or truly singular points where it blows up. This is the notion used here to design a global linearization atlas. In contrast, determining a *dynamically invariant linearization* which defines a complete set of generalized Koopman eigendistributions requires determining all distinct dynamically invariant sets on the manifold, which is clearly a much more challenging problem. While the Koopman eigenproblem can always be solved for integrable vector fields, it is generically intractable for chaotic dynamics, which is the most common setting where one would like to design a linearization atlas.

The question of how to design an optimal linearization atlas with a minimal number of charts is an important question. It is possible that the overall number of Koopman linearization charts that are needed may be quite small, perhaps as small as the number of fixed points of the vector field, $\text{Fix}(V)$. Due to the *Poincaré-Hopf theorem*, this number must be greater than zero if the Euler characteristic of the manifold does not vanish.

However, another bound on this number is given by the minimum number of open charts needed to cover a manifold, M , which is determined by the Lustnerik-Schnirelman category⁸⁰, $\text{cat}(M) + 1$. For example, only one set is needed for a contractible space, two for the n -sphere, S^n , and $n + 1$ for the n -torus, $(\times S^1)^n$. Ostrand's theorem⁸¹ on covering dimension implies that an upper bound on the number of charts needed is $\dim(M) + 1$, so that $\text{cat}(M) \leq \dim(M)$.

However, due to the requirement of analyticity, the number of Carleman linearization charts needed could still be larger than this. Hence, in this work, we will not attempt to find an optimal nor minimal Carleman atlas, i.e. an atlas that only requires the minimum number of Carleman charts for a given vector field necessary to cover a given manifold. Instead, we will seek to find robust numerical procedures for ensuring that the global Carleman embedding is convergent. We leave it as an open question to find an explicit algorithm for determining an optimal and/or minimal Carleman atlas.

E. Main results

We propose a globalized Carleman linearization method that enables the simulation of dynamical systems with multiple fixed points, including chaotic dynamical systems, and that allows one to track trajectories far from any fixed points, including those diverging from unstable equilibria. This technique divides the space into charts, each with a convergence radius ξ . If enough information is given about the

vector field, then the required atlas of charts can be precomputed before integrating the trajectories in time. Here, one constructs an atlas of linearization charts that covers the entire solution manifold and allows the Carleman embedding to remain convergent over the entire manifold.

Alternatively, linearization charts can be constructed dynamically along the trajectory. Here, the goal is only to generate a series of charts that cover a neighborhood of the trajectory of interest. The simple piecewise embedding method, defined below, follows the trajectory by transitioning from one chart to the next each time the trajectory reaches the boundary of the current chart. We also propose an adaptive version of the piecewise embedding, that varies the linearization chart radius ξ on the fly depending on the difference between two trajectories simulated within charts of different sizes. In this case, a larger or smaller radius is selected depending on whether a rapid decline in convergence is detected.

This paper is organized as follows. In Sec. II A, we review the standard Carleman technique. In Sec. II B, we extend the technique to a globalized piecewise adaptive version. In Sec. II C, we propose an alternative globalized embedding method with a static predefined atlas. In Sec. III, the piecewise technique is tested on various nonlinear problems progressing from 1D and 2D systems, where good decompositions are known, to 3D systems that harbor completely chaotic dynamics. For example, we study the ability to capture dynamics on strange attractors, such as the Rössler, Lorenz, and Chen's attractor. We also study integrable and non-integrable Lotka-Volterra predator-prey systems. Finally, Sec. IV concludes with a discussion of the performance of the globalized Carleman embedding algorithm and the prospects for developing a globalized quantum Carleman algorithm for use on future quantum computers.

II. GLOBALIZED CARLEMAN EMBEDDING

A. Standard Carleman embedding (SCE) method

Let us assume that we need to solve a set of nonlinear differential equations. Assume the solution $X = \{X^i\} \in \mathbb{R}^n$, is specified in terms of coordinates X^i and that the flow is specified by a nonlinear vector field $V(X) = \{V^i(X^j)\} \in \mathbb{R}^n$ via:

$$d_t X = V(X). \quad (1)$$

To construct the Carleman embedding, we Taylor expand the vector field $V(X)$ around a chosen point X_0 . In terms of $x = X - X_0$, this leads to the result:

$$d_t x = F(x) \simeq F_0 + F_1 x + F_2 x^{\otimes 2} + \dots F_p x^{\otimes p}. \quad (2)$$

When the nonlinearity is carried to order p , the resulting linear differential equation has order p . The vector $x \in \mathbb{R}^n$ is

$$x = (x^{(0)}, x^{(1)}, \dots, x^{(n-1)})^T, \quad (3)$$

where $x^{(j)}$ is the j -th component of the vector x . The vector $x^{\otimes 2}$ is a tensor product of two vectors x , i.e., $x \otimes x \in \mathbb{R}^{n^2}$, and so on. The matrices $F_k \in \mathbb{R}^{n \times n^k}$ describe the nonlinear problem in a given linearization chart with a convergence radius ξ :

$$|x| \leq \xi \leq 1. \quad (4)$$

It should be kept in mind that the elements of the matrices F depend on the point X_0 around which the chart is constructed.

To implement the SCE, one considers P monomials $x, x^{\otimes 2}, \dots, x^{\otimes P}$ as independent components of the vector u :

$$u = (x, x^{\otimes 2}, \dots, x^{\otimes P})^T, \quad (5)$$

where P must be greater or equal than p . The size of the vector u is determined by the number of independent homogeneous polynomials up to degree P which is given by the binomial coefficient

$$N = C_n^{n+P} = C_P^{n+P} = \frac{(n+P)!}{n!P!} < \frac{(n+P)^{n+P}}{n^n P^P}. \quad (6)$$

One can show that N is bounded from above by the smaller of P^n and n^P . In fact, using Stirling's approximation, when $P \gg n$, as is the case for the low-dimensional dynamical systems studied here, N scales as $(P/n)^n$ which is exponential in dimension n , but in the opposite limit, $n \gg P$, this scales as $(n/P)^P$ which is exponential in the polynomial degree P . For large systems of ODEs, such as those resulting from the discretization of PDEs, this leads to a significant reduction in degrees of freedom (DOFs) relative to tracking P polynomials for all n variables, which would require P^n terms.

The embedding results in the linear differential equation

$$d_t u = Au + B, \quad (7)$$

where the matrices $A \in \mathbb{R}^{N \times N}$ and $B \in \mathbb{R}^{N \times 1}$ include the matrices F_k as subblocks. For instance, for $p = 3$, the matrix A is

$$A = \begin{bmatrix} A_1^{[1]} & A_2^{[1]} & A_3^{[1]} & 0 & 0 & 0 & \dots & 0 \\ A_1^{[2]} & A_2^{[2]} & A_3^{[2]} & A_4^{[2]} & 0 & 0 & \dots & 0 \\ 0 & A_2^{[3]} & A_3^{[3]} & A_4^{[3]} & A_5^{[3]} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & 0 & A_{p-1}^{[P]} & A_p^{[P]} \end{bmatrix}. \quad (8)$$

Here, $A_k^{[j]} \in \mathbb{R}^{n^j \times n^k}$ where $k, j = 1, 2, \dots, P$. The matrices $A_k^{[j]}$ can be computed in the following way:

$$A_{j+p}^{[j]} = F_{p+1} \otimes I_N^{\otimes j-1} + I_n \otimes F_{p+1} I_N^{\otimes j-2} + \dots + I_N^{\otimes j-1} \otimes F_{p+1}, \quad (9)$$

where $p = -1, 0, 1, \dots, p-1$, and I_n is the $n \times n$ unit matrix. The matrix B depends only on F_0 :

$$B = [F_0^T, 0, \dots, 0]^T. \quad (10)$$

This large system of linear differential equations (7) embeds the nonlinear system (2). However, as explained above, the embedding (7) only works correctly within a subset of the linearization chart (4) and only in the vicinity of a single fixed point.

TABLE I: Piecewise adaptive Carleman embedding algorithm.

Given data
Time interval $t_{\max}$, coefficients c , initial conditions X_0 , function $\mathcal{F}_c$, function $\mathcal{S}_c$, tolerance ϵ_{tol} , initial chart size ξ , step $\Delta\xi$, minimal possible radius $\xi_{\min}$, maximum possible radius $\xi_{\max}$.
Embedding subroutine: $x(t + \Delta t)$, $\text{isUnstable} = \mathcal{E}(t, x_{\text{init}}, \Xi)$
$c' = \mathcal{F}_c(\Xi)$ $A, B, u_{\text{init}} = \mathcal{S}_c(c', x_{\text{init}})$ // -- A single time step of a Runge-Kutta solver -- $u = \text{RK}(A, B, u_{\text{init}}, t)$ $x = \mathcal{D}(u)$ $\text{isUnstable} = \text{any}(u > 1)$ return x , isUnstable
Adaptation subroutine: $t, x([t_{\text{init}}, t]), \xi_{\text{new}}, \Xi = \mathcal{A}(t_{\text{init}}, \xi, \Xi, \text{flagInc})$
$\xi' = \xi - \Delta\xi$ $x_{\text{arr}}, t, x, i_t = [0], t_{\text{init}}, 0, 0$ if $\xi' \geq \xi_{\min}$: // -- Simulate the reduced chart -- while $ x < \xi'$: $x, \text{isUnstable} = \mathcal{E}(t, x, \Xi)$ $i_t = i_t + 1$ $x_{\text{arr}}[i_t], t = x, t + \Delta t$ // -- Compare the reference and shifted reduced charts -- $\Xi', x' = \Xi + x, 0$ While $ x < \xi$: $\epsilon = x - x' + \Xi - \Xi'$ if $ \epsilon \geq \epsilon_{\text{tol}}$ or isUnstable : return $\mathcal{A}(t_{\text{init}}, \xi', \Xi, \text{false})$ // reduce the chart else: $x, \text{isUnstable} = \mathcal{E}(t, x, \Xi)$ $x', \Xi = \mathcal{E}(t, x', \Xi')$ $x_{\text{arr}}[i_t], t = x, t + \Delta t$ else: // -- Simulate just the reference chart -- while $ x < \xi$: $x, \Xi = \mathcal{E}(t, x, \Xi)$ $i_t = i_t + 1$ $x_{\text{arr}}[i_t], t = x, t + \Delta t$ if not flagInc: return $t, x_{\text{arr}}, \xi, \Xi + x_{\text{arr}}[i_t]$ // take the reference chart // -- Compare the reference and enlarged charts -- $\xi' = \xi + \Delta\xi$ While $\xi' \leq \xi_{\max}$: $\Xi', x' = \Xi + x, 0$ $x, \text{isUnstable} = \mathcal{E}(t, x, \Xi)$ // evolution in the enlarged chart $x', \Xi = \mathcal{E}(t, x', \Xi')$ // evolution in the shifted reference chart While $ x < \xi'$: $\epsilon = x - x' + \Xi - \Xi'$ if $ \epsilon \geq \epsilon_{\text{tol}}$ or isUnstable : return $t, x_{\text{arr}}, \xi, \Xi + x_{\text{arr}}[i_t]$ // take the reference chart else: $x_{\text{arr}}[i_t], t = x, t + \Delta t$ $x, \text{isUnstable} = \mathcal{E}(t, x, \Xi)$ $x', \Xi = \mathcal{E}(t, x', \Xi')$ $\xi = \xi'$ // the enlarged chart becomes the new reference chart $\xi' = \xi + \Delta\xi$ return $t, x_{\text{arr}}, \xi, \Xi + x_{\text{arr}}[i_t]$ // take the enlarged chart
Overall algorithm
$\Xi, t, X = X_0, 0, []$ While $t \leq t_{\max}$: $t_{\text{init}}, \Xi_{\text{init}} = t, \Xi$ $t, x([t_{\text{init}}, t]), \xi, \Xi = \mathcal{A}(t_{\text{init}}, \xi, \Xi_{\text{init}}, \text{true})$ $X([t_{\text{init}}, t]) = x([t_{\text{init}}, t]) + \Xi_{\text{init}}$ return X

B. Piecewise Adaptive Carleman Embedding (ACE)

The goal of this work is to extend the Carleman embedding method to a much larger class of nonlinear systems, including chaotic dynamical systems. In particular, to bypass the limitation (4), we allow the choice of embedding chart to change along the trajectory, as schematically shown in Fig. 1. First of all, we construct an embedding chart C_0 (green circle) centered at the initial conditions X_0 . The Car-

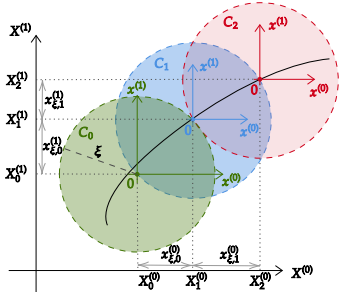


FIG. 1: Schematic depiction of the dynamic Piecewise Carleman Embedding (PCE) for a system with $n = 2$ degrees of freedom. A new linearization chart must be chosen each time the trajectory leaves the original linearization chart.

leman method evolves the trajectory (solid black line) until it reaches the chart boundary, beyond which the condition (4) is no longer satisfied. At this moment, one switches to a new chart, C_1 (blue circle), centered around the new point $X_1 = (X_1^{(0)}, X_1^{(1)})$ and rewrites the embedding (7). The displacement $x_{\xi,0}$ between two centers X_0 and X_1 is added to the global shift Ξ ,

$$\Xi = \Xi + x_{\xi}, \quad (11)$$

where Ξ is initialized as the zero vector. Later, the vector Ξ is used to reconstruct the original variable X from the local variable x via:

$$X = x + \Xi. \quad (12)$$

When the trajectory reaches the boundary of C_1 , one again jumps to the new chart C_2 (red circle), adds the local shift $x_{\xi,1}$ to Ξ , reconstructs the linearization (7) at the new point X_2 , and models the system's evolution within the chart C_2 , and so on.

This approach requires minimal prior knowledge about the simulated system. In particular, it does not require prior knowledge about the positions of the system's fixed points. However, it does require knowing the explicit dependence of the linearization (2) on the point X_j around which the chart C_j is constructed. For example, if c represents the set of coefficients characterizing the nonlinear system (2), a function $\mathcal{F}_c(\Xi)$ can be introduced to describe the dependence of c on the accumulated vector shift Ξ . The implementation of the function $\mathcal{F}_c(\Xi)$ strongly depends on the simulated system (1). Another function, $\mathcal{S}_c(c)$, constructs the embedding (7) using the coefficients c . The implementation $\mathcal{S}_c(c)$ follows the logic described in Sec. II A. The function $\mathcal{S}_c(c)$ takes the coefficients c , constructs the matrices F , Eq. (2), and then builds the embedding system (7) by preparing the matrices A and B , and the initial vector u . Once the trajectory within a single chart has been successfully simulated, e.g., by using a Runge-Kutta time integrator, a function $\mathcal{D}(u)$ is used to decode the embedding monomials u back into the local variable x . As seen from Eq. (5), $\mathcal{D}(u)$ can be implemented just by taking the first n elements of u . After this, the original variable X is reconstructed using Eq. (12).

One may still encounter an unfavorable scenario in which two or more fixed points lie within a single linearization chart, i.e., within the domain (4). To address this case, an adaptive method that dynamically adjusts the radius of each chart, ξ , is required. A schematic depiction of the logical flow diagram for an adaptive algorithm for dynamically adjusting the radius on the fly is shown in Fig. 2.

One starts from a reference chart with a radius ξ (black dashed circle in the case C_A) and compares the embedding with the linearization within a smaller chart, $\xi' = \xi - \Delta\xi$ (red dashed circle). Inside the ξ' -chart, the trajectories obtained by both linearizations coincide but can diverge outside the ξ' -chart starting from the point $X = O''$. Thus, one first simulates the trajectory within ξ' -chart, then shifts the chart to the point O'' and compares the reference trajectory with the trajectory in the shifted ξ' -chart at each time step. If necessary, the reduced chart must be shifted again when the simulated trajectory is found to lie outside the current reduced chart.

The difference between two trajectories is computed as

$$\varepsilon = |X - X'|, \quad (13)$$

where X is a point on the trajectory obtained using the ξ -chart, and X' is a point on the trajectory obtained using the shifted ξ' -embedding. It is important to note that the comparison is made between the global variables, rather than their local counterparts x and x' , since the local variables in the ξ - and ξ' -charts differ completely.

If ε is less than the required tolerance ε_{tol} for all X in the interval between O'' and O' , one saves the computed part of the trajectory. Then, one compares the reference linearization with the embedding in a larger chart, $\xi' = \xi + \Delta\xi$ (red circle in the case C_A^+). The linearizations coincide within the ξ -chart, but can diverge beyond the point O' . To perform the comparison, one needs to shift the reference chart to the point O' . If $\varepsilon < \varepsilon_{\text{tol}}$ for all points between O' and O^+ , the larger chart is taken as the reference, the trajectory $O'O''$ is saved, and the new chart is then compared with an even larger chart. The linearization radius is continuously increased as long as $\varepsilon < \varepsilon_{\text{tol}}$ and $\xi \leq \xi_{\text{max}} \leq 1$. Once $\varepsilon \geq \varepsilon_{\text{tol}}$ at some point $X = O^*$ (the blue dot in the case C_A^+), the step increase is halted, and the reference chart is shifted to the point O^* without increasing the convergence radius ξ . The comparison with larger charts is necessary to accelerate simulations by avoiding the use of over-reduced charts.

If one arrives back to the case C_A , there is a point $X = O^*$ in the interval between the points O'' and O' where $\varepsilon \geq \varepsilon_{\text{tol}}$, then the linearization radius is decreased, i.e. $\xi^* = \xi - \Delta\xi$. The ξ^* -chart is now taken as the reference one (the case C_A^- in Fig. 2). This chart is then compared with an even smaller chart $\xi^* - \Delta\xi$. If $\varepsilon \geq \varepsilon_{\text{tol}}$ is still observed, one keeps decreasing the linearization radius as long as ξ remains larger or equal than an imposed minimal value ξ_{min} . On the other hand, if $\varepsilon < \varepsilon_{\text{tol}}$, the ξ^* -embedding is kept as the reference, the trajectory OO' is saved, and the chart is shifted to the point O' (the case C_A').

Hence, the original chart C_A is shifted either to the chart C_A' or the chart C_A^* . After the shift, the algorithm follows the same logic as it did for the chart C_A .

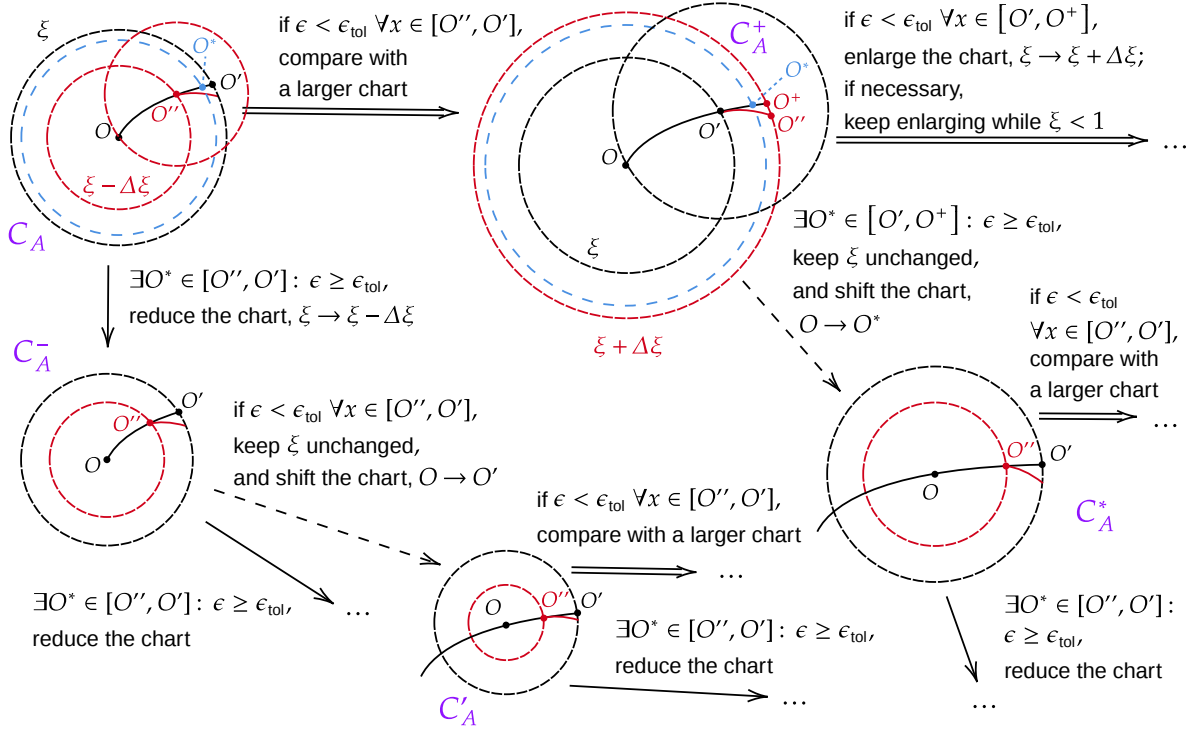


FIG. 2: The logical flow diagram for the ACE algorithm that adapts the new chart radius ξ based on the need to satisfy an imposed error threshold ϵ_{tol} .

It is worth noting that the chart size can be adjusted either by using an additive step $\Delta\xi$, or, for example, by multiplying or dividing ξ by a factor of two to accelerate changes in the radius. These two approaches can be combined: use the step $\Delta\xi$ when $\xi \geq \xi_{\text{th}}$ and apply multiplication or division when $\xi < \xi_{\text{th}}$ where ξ_{th} is a preset value.

The described piecewise Adaptive Carleman Embedding (ACE) is summarized in Table I. We also distinguish its simplified non-adaptive version, i.e., non-adaptive Piecewise Carleman Embedding (PCE). In the latter algorithm, one still moves the linearization chart along the simulated trajectory but keeps the convergence radius ξ constant.

The current version of the adaptive method depends on the step size $\Delta\xi$ used to adjust the convergence radius. To eliminate this dependence, the logic of the adaptive scheme can be reversed: starting from a small value of ξ , the convergence radius is progressively increased by applying a multiplicative factor $\eta > 1$, such that $\xi \rightarrow \eta\xi \rightarrow \eta^2\xi \rightarrow \dots$. Once $\epsilon \geq \epsilon_{\text{tol}}$, the convergence radius is reduced by a multiplicative factor $1/\eta'$, where $\eta' = \eta/\gamma$ and $\gamma > 1$ is an irrational number. If, after this radius reduction, i.e., $\xi \rightarrow \xi/\eta'$, the condition $\epsilon < \epsilon_{\text{tol}}$ is satisfied, the radius is increased again using a modified factor $\eta'' = \eta/\gamma^2$, and this process continues accordingly. By using this technique, it is potentially possible to eliminate the dependence on $\Delta\xi$ and to determine a more optimal value of ξ for each segment of the simulated trajectory. The drawback of this approach is that the number of ξ adjustments could become substantial.

C. Grid-anchored static piecewise Carleman Embedding (GCE)

The adaptive embedding approach described in the previous section requires not only detecting the moment when the linearization chart must be switched but also comparing trajectories across charts of different sizes, or employing another method for detecting the Carleman linearization instability within a given chart. Even the simplified PCE version, which lacks adaptivity in chart radius, results in dynamically moving linearization regions that requires high computational overhead. These adaptive methods, which rely on many iterative nonlinear operations, cannot be implemented on a quantum computer in a manner that achieves a quantum advantage⁶.

Therefore, we propose an alternative version of the embedding using static linearization charts, which we refer to as elements or tiles. Specifically, we perform clustering of the evolution space of the variables $X^{(i)}$, with $i = 0, 1, \dots, n-1$, prior to executing the Carleman simulations. This results in a static grid of linearization regions, Fig. 3, whose positions are known in advance. This grid consists of elements with fixed sizes $2\xi^{(i)}$ in the i -th dimension and must satisfy the condition

$$|\xi| \equiv \sqrt{\sum_{i=0}^{n-1} (\xi^{(i)})^2} \leq 1, \quad (14)$$

to ensure the convergence of the Carleman linearization near the corners of the tiles. If the tile sizes are equal in all direc-

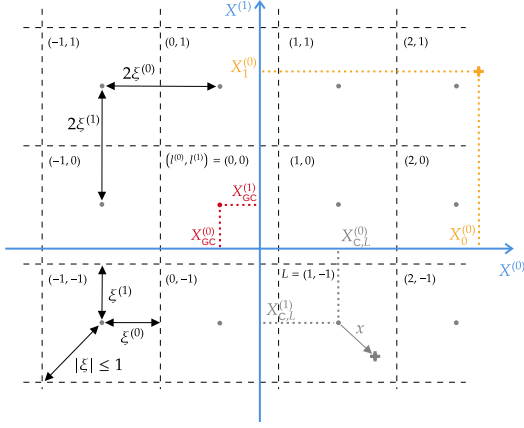


FIG. 3: Schematic depiction of the static piecewise Grid-anchored Carleman Embedding (GCE) for the case with $n = 2$ variables. The original nonlinear variables $X^{(i)}$ are shown in blue. The red dot at $(X_{GC}^{(0)}, X_{GC}^{(1)})$ marks the center of the linearization grid. The position of each tile-shaped element is determined by the integers $l^{(i)}$. The distance between the centers of two neighboring elements is $2\xi^{(i)}$ in the i -th direction. The orange cross indicates the location of the initial condition, $(X_0^{(0)}, X_0^{(1)})$.

tions than the above condition can be recast as

$$\xi^{(i)} \leq \frac{1}{\sqrt{n}}. \quad (15)$$

The center of the linearization grid (the red dot in Fig. 3) is placed at the position

$$X_{GC} = (X_{GC}^{(0)}, X_{GC}^{(1)}, \dots, X_{GC}^{(n-1)})^T, \quad (16)$$

which does not necessarily have to be close to the initial condition (the orange cross in Fig. 3) of the simulated system of nonlinear equations. The positions of the tile elements are described by the integer indices $l^{(i)}$, with $i = 0, 1, \dots, n-1$, where $l^{(i)} = 0 \forall i$ marks the center tile of the linearization grid. The center of the tile associated with the integer vector

$$L = (l^{(0)}, l^{(1)}, \dots, l^{(n-1)})^T, \quad (17)$$

is computed as

$$X_{C,L}^{(i)} = X_{GC}^{(i)} + 2\xi^{(i)} l^{(i)}, \quad i = 0, 1, \dots, n-1, \quad (18)$$

or, equivalently, in vector form

$$X_{C,L} = X_{GC} + 2 \text{diag}(\xi) L. \quad (19)$$

Within each linearization chart, we work with a local variable x , whose components satisfy the condition

$$|x^{(i)}| \leq \xi^{(i)}, \quad i = 0, 1, \dots, n-1, \quad (20)$$

similarly to the local variables introduced in Sec. II A. The local variable is equal to zero at the center of each tile. The original variable X in the L -th tile can be expressed as the sum of the local variable x and the center of the corresponding tile:

$$X = X_{C,L} + x. \quad (21)$$

Thus, similarly to the global shift Ξ introduced in Sec. II B, the tile center $X_{C,L}$ is used to construct the local Carleman embedding:

$$\Xi \equiv X_{C,L}. \quad (22)$$

Knowing $X_{C,L}$, one can derive the coefficients c' , which describe the nonlinear system in the L -th chart,

$$c' = \mathcal{F}_c(X_{C,L}), \quad (23)$$

from which the Carleman embedding (7) is derived:

$$A, B, u_{\text{init}} = \mathcal{S}_c(c', x_{\text{init}}). \quad (24)$$

This is similar to the embedding subroutine used in Table I.

To initialize the Carleman embedding on the static linearization grid, one computes the integer vector L_0 , which indicates the tile containing the initial condition. The components of the vector L_0 are computed as

$$l_0^{(i)} = \left\lfloor \frac{X_0^{(i)} - X_{GC}^{(i)}}{2\xi^{(i)}} \right\rfloor. \quad (25)$$

After this, the center of this tile, X_{C,L_0} , can be found using Eq. (19). The local coordinate of the initial condition is

$$x_{\text{init},0} = X_0 - X_{C,L_0}. \quad (26)$$

Using X_{C,L_0} and $x_{\text{init},0}$, one constructs the initial Carleman embedding by employing Eqs. (23) and (24). Then, the vector x is evolved in time, and when the local coordinates no longer satisfy condition (20), the system transitions to a neighboring chart. In particular, for each direction k such that

$$|x^{(k)}| > \xi^{(k)}, \quad (27)$$

the components $l^{(k)}$ of the vector L are modified as

$$l^{(k)} \rightarrow l^{(k)} + \frac{x^{(k)}}{|x^{(k)}|}. \quad (28)$$

Then, one computes the center of the new tile using Eq. (19) and constructs new local embedding using Eqs. (23) and (24), and so on.

In general, the geometry of the charts can be adapted to the topology of the system's evolution to minimize the number of shifts between linearization regions. However, in this case, detecting the appropriate moment of the transition between neighboring charts can be more complicated than in the case of the hypercubes used in Fig. 3. In principle, the construction of the linearization grid should account for the

number and positions of fixed points in the nonlinear system under consideration, in order to ensure the stability of the Carleman embedding and to minimize the number of tile transitions. The static grid can be scaled by adjusting ξ or shifted by modifying X_{GC} to adapt to a particular nonlinear system. For instance, each tile should include only a single fixed point. Moreover, one should avoid placing a fixed point, such as the center of a limit cycle, near the corner of multiple tiles. Otherwise, the number of tile transitions will be significant when the system evolves near that fixed point. Typically, the accuracy of the linearization deteriorates near the chart boundaries. For this reason, it is advisable to avoid scenarios where the trajectory stays close to the boundary for an extended period, especially when using large tiles. Otherwise, numerical errors may accumulate.

Since the position and the parameters of the anchored grid are known in advance, the described GCE technique may be more suitable for applications that require great speed. However, one still needs to detect the moment when chart transitions are necessary. In particular, this operation is problematic for quantum computers and, most likely, can only be implemented via measurement. In the worst case, all components of the local vector x must be analyzed to detect the precise moment when a chart transition becomes necessary. This potentially requires measuring n values, i.e. all n components of x . This situation is similar to the PCE technique, where one must measure all n components of the vector x_ξ to construct a new embedding in the dynamically shifted chart.

III. NUMERICAL TESTS

In this section, we evaluate the globalized Carleman methods proposed in the previous section on a set of nonlinear test problems and compare the results with standard classical (CL) simulations, where the nonlinear dynamical system is directly integrated in time using a Runge-Kutta method. To compare CL and Carleman simulations, we compute the relative error

$$\varepsilon_{\text{num}}(X^{(i)}) = \frac{|X_{\text{CL}}^{(i)} - X_{\text{CE}}^{(i)}|}{|X|}, \quad (29)$$

where $X_{\text{CL}}^{(i)}$ is the i -th variable computed by the CL simulation, $X_{\text{CE}}^{(i)}$ is the i -th variable computed by the PCE, ACE, or GCE techniques, and $|X|$ is

$$|X| = \sqrt{\sum_{i=0}^{n-1} |X_i|^2}. \quad (30)$$

The implementation of the PCE, ACE, and GCE techniques, along with the numerical simulations presented in this section, are provided in Ref. 82.

A. 1D Systems: One degree of freedom

To test the PCE and ACE methods, we first simulate a simple nonlinear problem with $p = 3$ and a single scalar variable

X :

$$d_t X = (X_{c,1} - X)(X_{c,2} - X)(X_{c,3} - X), \quad (31)$$

where the coefficients $X_{c,1}$, $X_{c,2}$, and $X_{c,3}$ indicate the positions of the fixed points. One can see the simulation results in Fig. 4 for several sets of $X_{c,k}$. In particular, the plots (a) and (b) show standard Carleman simulations (SCE, black dashed lines). The SCE simulations are unstable and cannot reproduce the evolutionary CL trajectories (gray lines). On the other hand, the PCE simulations (red dashed lines) with $\xi = 0.1$ and $P = 6$ correctly model the nonlinear problems and can catch all fixed points in the system starting from different initial conditions.

The comparison between PCE and ACE techniques is demonstrated in plots (c) and (d). In both cases, if ξ is not small enough, the PCE technique can either have relatively high error or fail to converge to the correct fixed point (blue dashed line in (d)). ACE also does not find an optimal ξ if the tolerance ε_{tol} is not sufficiently small; for example, see green line in (c) or red line in (d). In particular, in the case with $\xi_{\text{max}} = 1.0$ and $\varepsilon_{\text{tol}} = 10^{-4}$, i.e., the green line in (c), ACE reduces the chart radius only down to $\xi \approx 0.87$. This results in high numerical errors.

To improve the ACE precision, one can either impose a smaller ξ_{max} or set a lower tolerance ε_{tol} . For instance, if the tolerance is decreased from 10^{-4} to 10^{-10} (black line in (c)), ACE reduces ξ down to ≤ 0.12 (the chart radius varies in time for different parts of the simulated trajectory), which results in significantly lower numerical errors.

One can also increase $\Delta\xi$. When $\Delta\xi$ is small, then two charts being compared are nearly the same size, leading to only minor differences between the trajectories simulated within them. Thus, a sufficiently low tolerance must be imposed to ensure that the ACE algorithm is sensitive to differences between the trajectories. If the tolerance is not small enough, then the ACE algorithm may treat the charts with ξ and $\xi - \Delta\xi$ as equivalent, resulting in no update to ξ along the trajectory. By increasing $\Delta\xi$, the difference between the two trajectories becomes more noticeable, allowing a higher tolerance to be used without reducing the maximum allowable convergence radius ξ_{max} . This effect is demonstrated in the numerical simulations presented below.

B. 2D Limit Cycles: the Van der Pol and Lotka-Volterra predator-prey models

We now test the PCE and ACE techniques on two-dimensional nonlinear systems, including the Van der Pol limit cycle and the Lotka-Volterra predator-prey model. To simulate the Van der Pol limit cycle,

$$d_t X = Y, \quad (32a)$$

$$d_t Y = \mu(1 - X^2)Y - X, \quad (32b)$$

we introduce new variables

$$x = X - \xi_x, \quad (33a)$$

$$y = Y - \xi_y. \quad (33b)$$

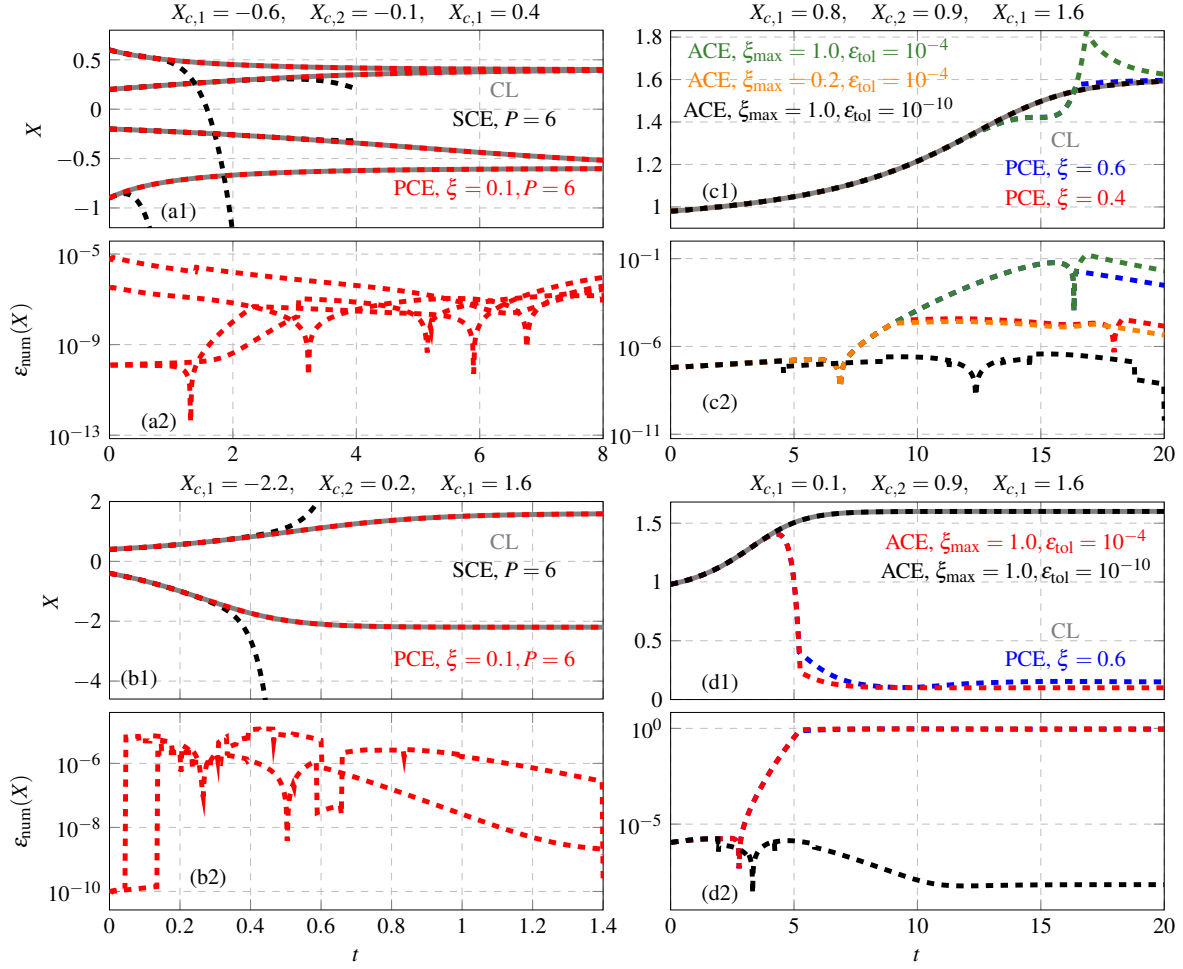


FIG. 4: Results from simulations of the problem (31) with $X_{c,1} = -0.6, X_{c,2} = -0.1, X_{c,3} = 0.4$ (a); $X_{c,1} = -2.2, X_{c,2} = 0.2, X_{c,3} = 1.6$ (b); $X_{c,1} = 0.8, X_{c,2} = 0.9, X_{c,3} = 1.6$ (c); $X_{c,1} = 0.1, X_{c,2} = 0.9, X_{c,3} = 1.6$ (d). The CL simulations are shown by gray lines. (a) and (b): Comparison with the SCE (black dashed lines) and PCE (red dashed lines) simulations for various initial conditions. (c) and (d): Comparison with the ACE and PCE simulations. Plots (a2), (b2), (c2), and (d2) shown numerical errors in the PCE and ACE simulations. All PCE and ACE simulations have $P = 6$. All ACE simulations have $\xi_{init} = \xi_{max}$ and $\Delta \xi = 0.02$.

After substituting Eqs. (33) into the system (32), one obtains the following system of equations for the variables x and y :

$$d_t x = y + \xi_y, \quad (34a)$$

$$d_t y = (-\xi_x + \mu \xi_y - \mu \xi_x^2 \xi_y) - (1 + 2\mu \xi_x \xi_y)x + \mu(1 - \xi_x^2)y - \mu \xi_y x^2 - 2\xi_x \mu xy - \mu x^2 y. \quad (34b)$$

The corresponding elements of the matrices F_p , defined in Eq. (2), are

$$F_{0,0} = \xi_y, \quad F_{0,1} = -\xi_x + \mu \xi_y - \mu \xi_x^2 \xi_y, \quad (35a)$$

$$F_{1,01} = 1, \quad (35b)$$

$$F_{1,10} = -(1 + 2\mu \xi_x \xi_y), \quad F_{1,11} = \mu(1 - \xi_x^2), \quad (35c)$$

$$F_{2,10} = -\mu \xi_y, \quad F_{2,11} = -2\xi_x \mu, \quad (35d)$$

$$F_{3,11} = -\mu, \quad (35e)$$

where $F_{p,rc}$ denotes the element in the r -th row and c -th column of the p -th matrix F_p , and F_0 is a vector. The variables

x and y are considered local within a particular chart. Thus, their values are restricted to the interval $[-\xi, \xi]$. Both variables are initialized to zero. Once $x^2 + y^2 \geq \xi^2$, the embedding chart is shifted, as illustrated in Fig. 1. The values of x and y taken immediately before the moment when the condition (4) is violated correspond to $x_{\xi}^{(0)}$ and $x_{\xi}^{(1)}$, respectively, according to the notation used in Fig. 1. The coefficients ξ_x and ξ_y accumulate the shifts in the linearization charts, $\Xi \equiv (\xi_x, \xi_y)$. More precisely, $\xi_x = \xi_y = 0$ for the initial linearization, i.e., within the initial chart C_0 , Fig. 1. When the chart shifts from C_0 to C_1 , the values $x_{\xi,0}^{(0)}$ and $x_{\xi,0}^{(1)}$ are added to ξ_x and ξ_y , respectively. Then, x and y are reset to zero, the embedding (7) is reconstructed using matrices (35) with the updated values of the coefficients ξ_x and ξ_y , and the system of linear equations (7) is evolved within the newly con-

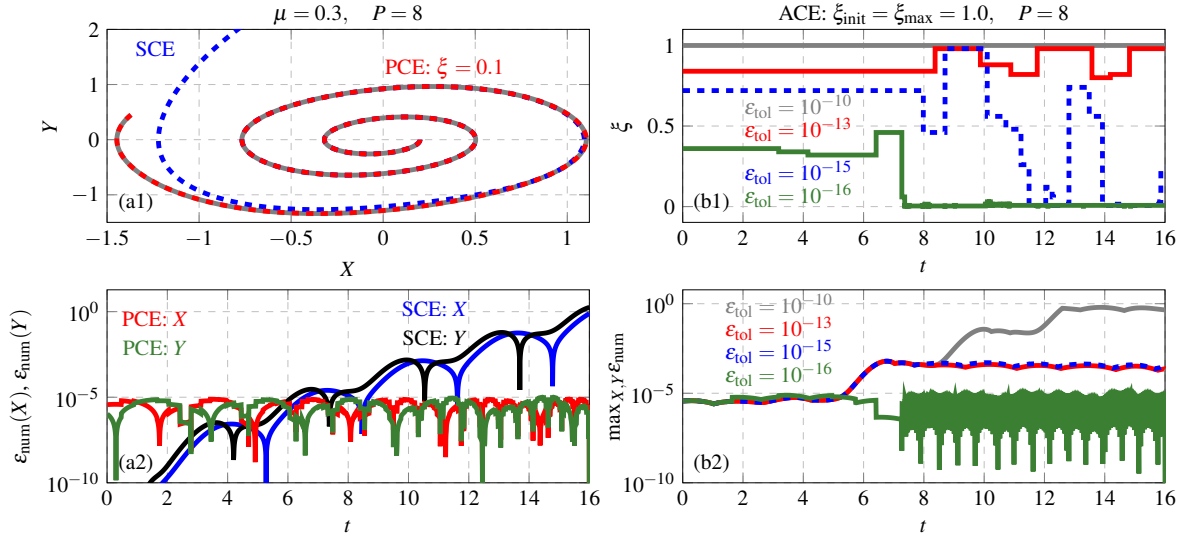


FIG. 5: Results of simulating the Van der Pol limit cycle (34) with the initial conditions $(X, Y) = (0.2, 0.0)$. (a1) SCE (blue dashed line) and PCE (red dashed line) simulations and the comparison with the CL simulation (gray line). (a2) Error of the SCE and PCE results. (b1) Evolution of the convergence radius ξ in the ACE simulations under various values of the tolerance and with a fixed step size of $\Delta\xi = 0.02$. (b2) Maximum error of the ACE results vs. time.

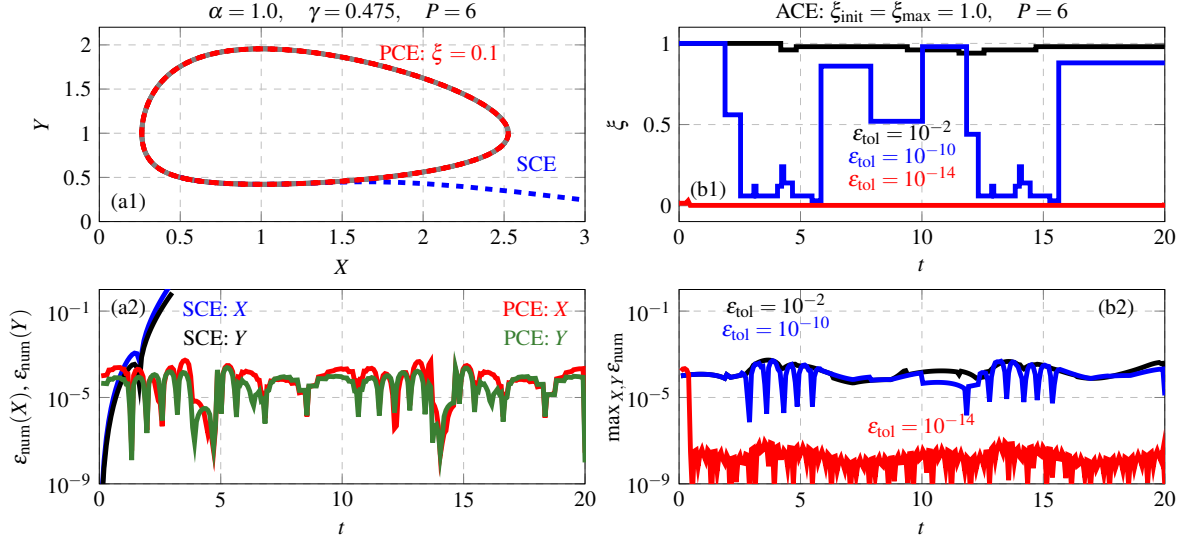


FIG. 6: Results of simulating the Lotka-Volterra predator-prey model (38) with the initial conditions $(X, Y) = (0.5, 0.5)$. (a1): SCE (blue dashed line) and PCE (red dashed line) simulations compared to the CL simulation (gray line). (a2): Error of the SCE and PCE results. (b1): Evolution of the radius ξ in the ACE simulations with various values of the tolerance and with a fixed step size of $\Delta\xi = 0.02$. (b2): Maximum error of the ACE results vs. time.

structured chart C_1 . After M chart shifts, the vector Ξ is

$$\Xi \equiv (\xi_x, \xi_y) = \left(\sum_{k=0}^{M-1} x_{\xi,k}^{(0)}, \sum_{k=0}^{M-1} x_{\xi,k}^{(1)} \right). \quad (36)$$

The vector Ξ is used to reconstruct the original variables X and Y using Eq. (12).

The results of modeling the Van der Pol limit cycle are presented in Fig. 5. The standard Carleman embedding (blue dashed line in (a1)) accurately reproduces the nonlinear dynamics for small values of X and Y , but becomes unstable as

the trajectory grows. In contrast, the PCE model (red dashed line in (a1)) successfully captures the full limit cycle dynamics.

We also applied the ACE technique to model the limit cycle under various tolerance values ϵ_{tol} , using a fixed step size of $\Delta\xi = 0.02$, as shown in plots (b). It can be observed that lowering the tolerance prompts the ACE method to adjust the convergence radius more actively, resulting in reduced numerical errors. However, this increased sensitivity comes at the cost of greater computational complexity, making the

ACE simulations more demanding as the tolerance is lowered.

Another interesting problem with two degrees of freedom is the Lotka-Volterra predator-prey model

$$d_t X = \alpha X - \beta XY, \quad (37a)$$

$$d_t Y = -\gamma Y + \delta XY. \quad (37b)$$

Here, we normalize the variables X and Y to γ/δ and α/β , correspondingly, and introduce the shifts (33). The resulting system becomes

$$d_t x = (\alpha \xi_x - \alpha \xi_x \xi_y) + (\alpha - \alpha \xi_y)x - \alpha \xi_x y - \alpha xy, \quad (38a)$$

$$d_t y = (-\gamma \xi_y + \gamma \xi_x \xi_y) + \gamma \xi_y x + (-\gamma + \gamma \xi_x)y + \gamma xy. \quad (38b)$$

According to system (38), the corresponding elements of the matrices F_p , introduced in Eq. (2), are

$$F_{0,0} = \alpha \xi_x - \alpha \xi_x \xi_y, \quad F_{0,1} = -\gamma \xi_y + \gamma \xi_x \xi_y, \quad (39a)$$

$$F_{1,00} = \alpha - \alpha \xi_y, \quad F_{1,10} = \gamma \xi_y, \quad (39b)$$

$$F_{1,01} = -\alpha \xi_x, \quad F_{1,11} = -\gamma + \gamma \xi_x, \quad (39c)$$

$$F_{2,01} = -\alpha, \quad F_{2,11} = \gamma. \quad (39d)$$

The comparison between the classical and Carleman simulations of the system (38) is shown in Fig. 6. One can see that, while the SCE simulation is unstable, the PCE and ACE techniques accurately reproduce the nonlinear dynamics of the predatory-prey model. The tradeoff of the how the choice of various parameters, such as the tolerance, affect the accuracy and complexity of the globalized algorithm are explored in the subpanels of the figure. If the error tolerance in the ACE simulation is not sufficiently small (black lines in (b)), ACE keeps the chart size close to the maximum possible value. Contrary to the case of the Van der Pol limit cycle, this does not result in large numerical errors. The ACE method with $\varepsilon_{\text{tol}} = 10^{-10}$ (blue lines) varies the linearization size ξ over a broad range of values: in some parts of the trajectory $\xi < 0.1$, while in others ξ is close to one. Nevertheless, the resulting numerical error in the ACE simulation is similar to that of the PCE simulation with $\xi = 0.1$. The ACE case with an even smaller tolerance (red lines) appears to overfit the nonlinear problem, resulting in much smaller but strongly oscillatory numerical errors. The same behavior is observed in the Van der Pol case with the tolerance $\varepsilon_{\text{tol}} = 10^{-16}$ (green lines in Fig. 5b).

C. The Lotka-Volterra superpredator-predator-prey model

To test the PCE and ACE techniques on more complicated models, we now simulate systems with three degrees of freedom. Let us consider the Lotka-Volterra superpredator-predator-prey (SPP) model with three independent variables:

$$d_t X = \alpha X - \beta XY, \quad (40a)$$

$$d_t Y = \varepsilon XY - \varepsilon YZ, \quad (40b)$$

$$d_t Z = -\eta Z + \eta Y. \quad (40c)$$

Similar to the previously considered cases, we introduce the shifts ξ_x , ξ_y , and ξ_z , which accumulate the displacements x_ξ according to Eq. (11). The system becomes

$$d_t x = (\alpha \xi_x - \beta \xi_x \xi_y) + (\alpha - \beta \xi_y)x - \beta \xi_x y - \beta xy, \quad (41a)$$

$$d_t y = (\varepsilon \xi_x \xi_y - \varepsilon \xi_y \xi_z) + \varepsilon \xi_y x + (\varepsilon \xi_x - \varepsilon \xi_z)y - \varepsilon \xi_y z + \varepsilon xy - \varepsilon yz, \quad (41b)$$

$$d_t z = (-\eta \xi_z + \eta \xi_y) + \eta y - \eta z. \quad (41c)$$

Results from simulating the three-dimensional Lotka-Volterra model are presented in Fig. 7. The standard embedding (blue line) becomes unstable very quickly, while the PCE technique with $\xi = 0.1$ successfully reproduces the nonlinear dynamics of the SPP model. Figures 7 also demonstrates that the ACE simulations with a step size of $\Delta \xi = 0.02$ require $\xi_{\text{max}} \leq 0.8$ to achieve the same level of precision as the PCE simulation (compare red and green lines in (b)). However, using a larger step size, $\Delta \xi = 0.2$ (orange line), allows ACE to detect variations in the evolutionary trajectory more easily and better adapt the convergence radius ξ . As a result, one can reduce the tolerance and use $\xi_{\text{max}} = 1$, while maintaining precision comparable to the PCE simulations with a small ξ . While the cases with $\Delta \xi = 0.02$ (red and green lines) keep the linearization size close to its maximum value, the case with $\Delta \xi = 0.2$ decreases ξ from 1.0 to 0.6. A similar effect in the case with $\Delta \xi = 0.02$ can be achieved by reducing the tolerance to $\varepsilon_{\text{tol}} = 10^{-9}$ (blue line). In all cases, ACE tends to decrease ξ over time, which can be attributed to the decreasing radius of the trajectory, requiring smaller charts for accurate linearization of the nonlinear problem.

D. The Rössler, Lorenz, and Chen attractors

We also consider chaotic systems such as the Rössler, Lorentz, and Chen strange attractor. The Rössler attractor is described by the following system of equations:

$$d_t X = -Y - Z, \quad (42a)$$

$$d_t Y = X + \sigma Y, \quad (42b)$$

$$d_t Z = \frac{\beta}{\eta \rho} - \rho Z + \eta \rho XZ. \quad (42c)$$

Introducing the shifts ξ_x , ξ_y , and ξ_z recasts the above system of equations as

$$d_t x = -(\xi_y + \xi_z) - y - z, \quad (43a)$$

$$d_t y = (\xi_x + \sigma \xi_y) + x + \sigma y, \quad (43b)$$

$$d_t z = \left(\frac{\beta}{\eta \rho} - \rho \xi_z + \eta \rho \xi_x \xi_z \right) + \eta \rho \xi_z x + (\eta \rho \xi_x - \rho)z + \eta \rho xz. \quad (43c)$$

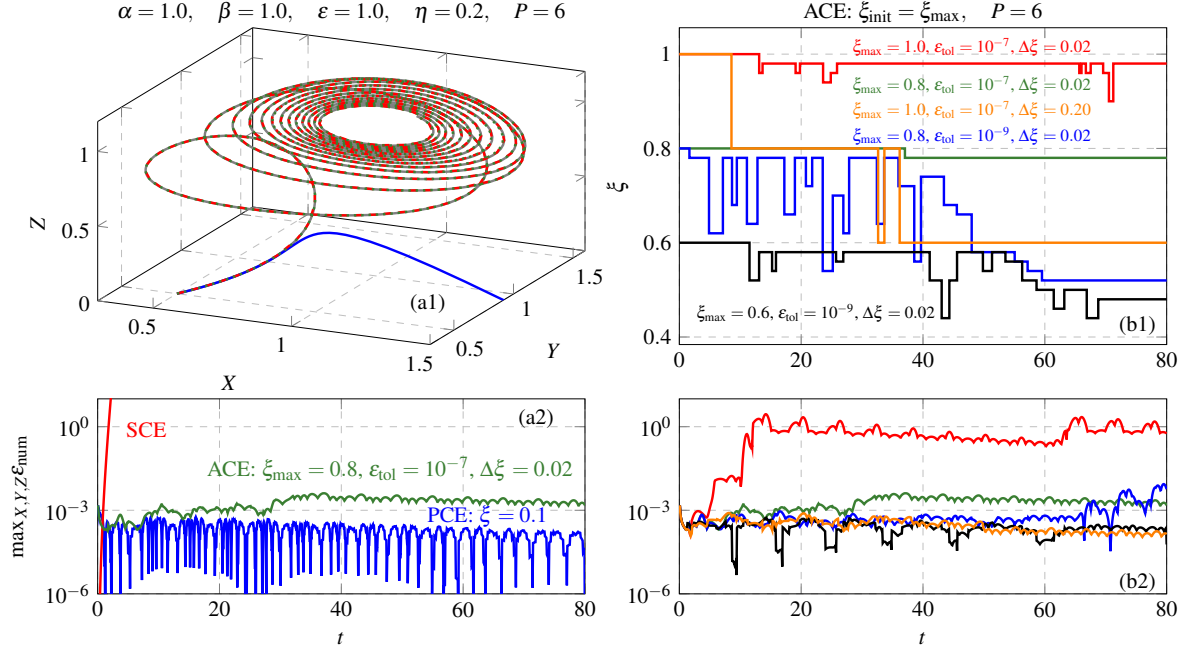


FIG. 7: (a1): Simulations of the Lotka-Volterra model (41) with three variables using CL (gray line), SCE (blue line), PCE (red dashed line), and ACE (green dashed line). The initial conditions are $(X, Y, Z) = (0.5, 0.5, 0.0)$. (a2): Maximum numerical errors in the PCE and ACE simulations. (b1) and (b2): Evolution of ξ and maximum numerical errors in the ACE simulations for various values of the tolerance, maximum convergence radius, and step size $\Delta\xi$.

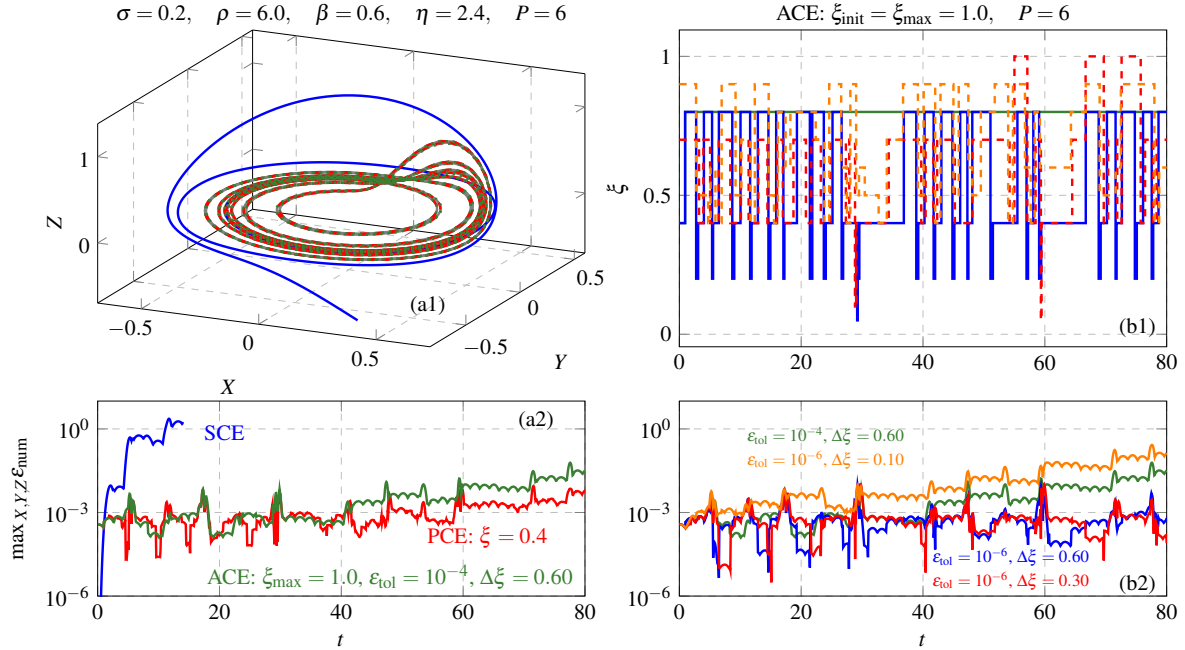


FIG. 8: (a1) Simulations of the Rössler attractor (43) using CL (gray line), SCE (blue line), PCE (red dashed line), and ACE (green dashed line). The initial conditions are $(X, Y, Z) = (0.0, 0.4, 0.0)$. (a2): Maximum numerical errors in the PCE and ACE simulations. (b1) and (b2): Evolution of ξ and maximum numerical errors in the ACE simulations for various values of the tolerance and step size $\Delta\xi$.

Results from modeling the Rössler attractor (43) are shown in Fig. 8. The SCE technique (blue line) fails to reproduce the nonlinear dynamics. The PCE method with a small ξ

and the ACE technique manage to successfully model the attractor. One can see from Fig. 8.b2 that, similarly to the previous cases, a low step size $\Delta\xi$ can result in higher nu-

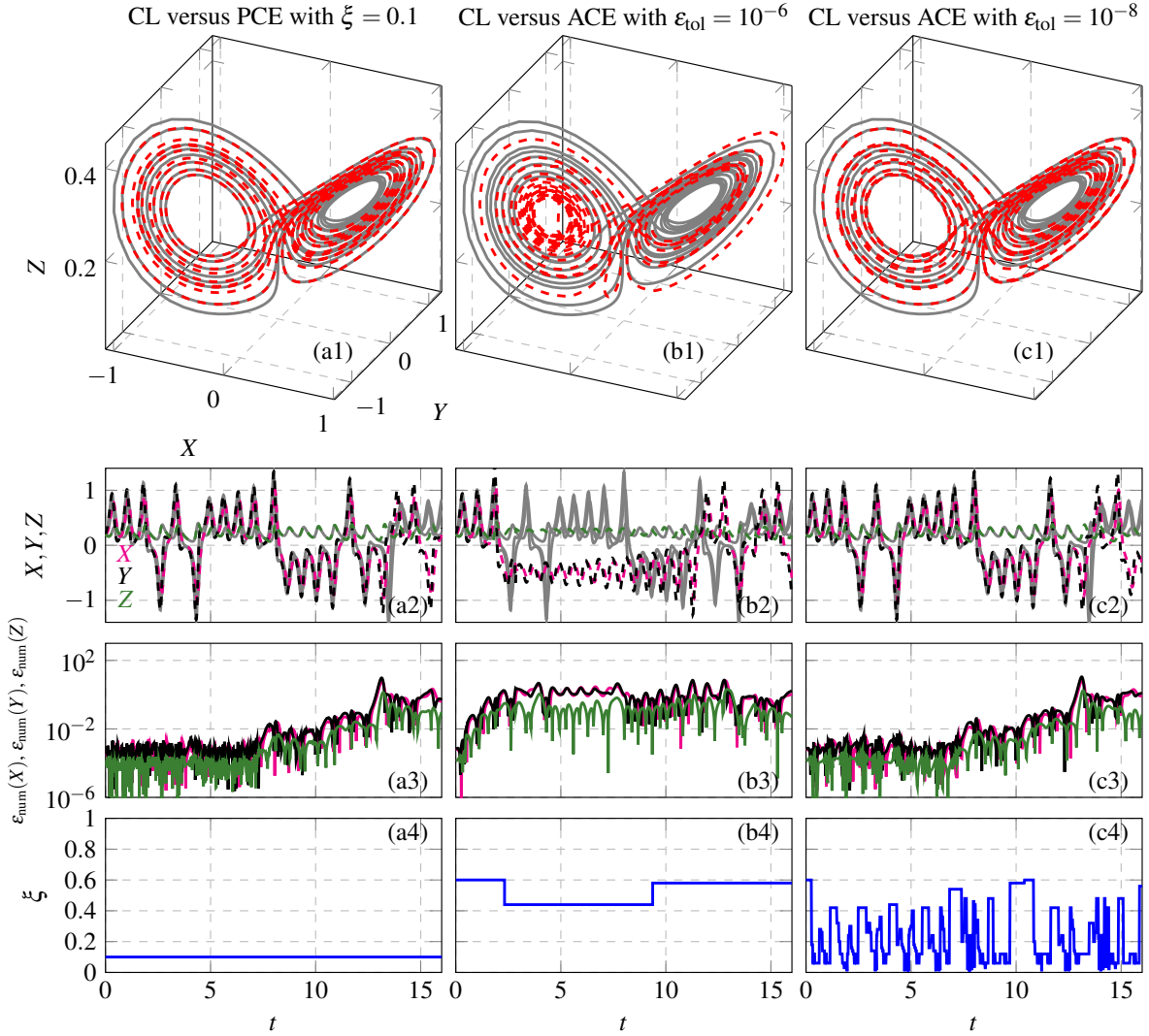


FIG. 9: Simulations of the Lorenz strange attractor (45) with $\sigma = 10$, $\rho = 28$, $\beta = 8/3$, and $c_z = 2c_x = 4$. The initial conditions are $(X, Y, Z) = (0.2, 0.2, 0.2)$. The gray lines are CL simulations. The colored lines are PCE (a) or ACE (b,c) simulations. All Carleman simulations used $P = 6$. All ACE simulations used $\xi_{\text{init}} = \xi_{\text{max}} = 0.6$ and $\Delta\xi = 0.02$. The bottom row presents the temporal evolution of ξ .

merical errors (for example, compare orange and red lines). In addition, ACE requires a sufficiently small tolerance for the chosen step $\Delta\xi$ to minimize the numerical error (compare the green and blue lines). The spikes in the numerical error observed in both PCE and ACE simulations occur during the time intervals when the trajectories rapidly grow in the z direction. ACE also appears to reduce ξ during these spikes (for example, compare the blue line in (b1) with that in (b2)). This may indicate that the crest in (a1) is the most demanding part of the attractor for linearization and requires the largest number of linearization regions.

To prepare the system of equations describing the Lorenz attractor

$$d_t X = \sigma(Y - X), \quad (44a)$$

$$d_t Y = X(\rho - Z) - Y, \quad (44b)$$

$$d_t Z = XY - \beta Z. \quad (44c)$$

so that they are suitable for the Carleman embedding, we normalize the variables X , Y , and Z to η_x , η_y , and η_z , respectively, where $\eta_x = \eta_y = c_x \sqrt{\beta(\rho - 1)}$ and $\eta_z = c_z(\rho - 1)$. We also introduce the shifts ξ_x , ξ_y , and ξ_z , as was done in all previous cases. As a result, the system is recast as:

$$d_t x = \sigma(\xi_y - \xi_x) - \sigma x + \sigma y, \quad (45a)$$

$$d_t y = (\rho \xi_x - \xi_y - \eta_z \xi_x \xi_z) + (\rho - \eta_z \xi_z)x - y - \eta_z \xi_x z - \eta_z x z, \quad (45b)$$

$$d_t z = (\eta \xi_x \xi_y - \beta \xi_z) + \eta \xi_y x + \eta \xi_x y - \beta z + \eta xy, \quad (45c)$$

where $\eta = \eta_x^2 / \eta_z$. The elements of the matrices F_p can be derived from the above system.

We also model the Chen attractor:

$$d_t X = \sigma(Y - X), \quad (46a)$$

$$d_t Y = (\rho - \sigma)X - XZ + \rho Y, \quad (46b)$$

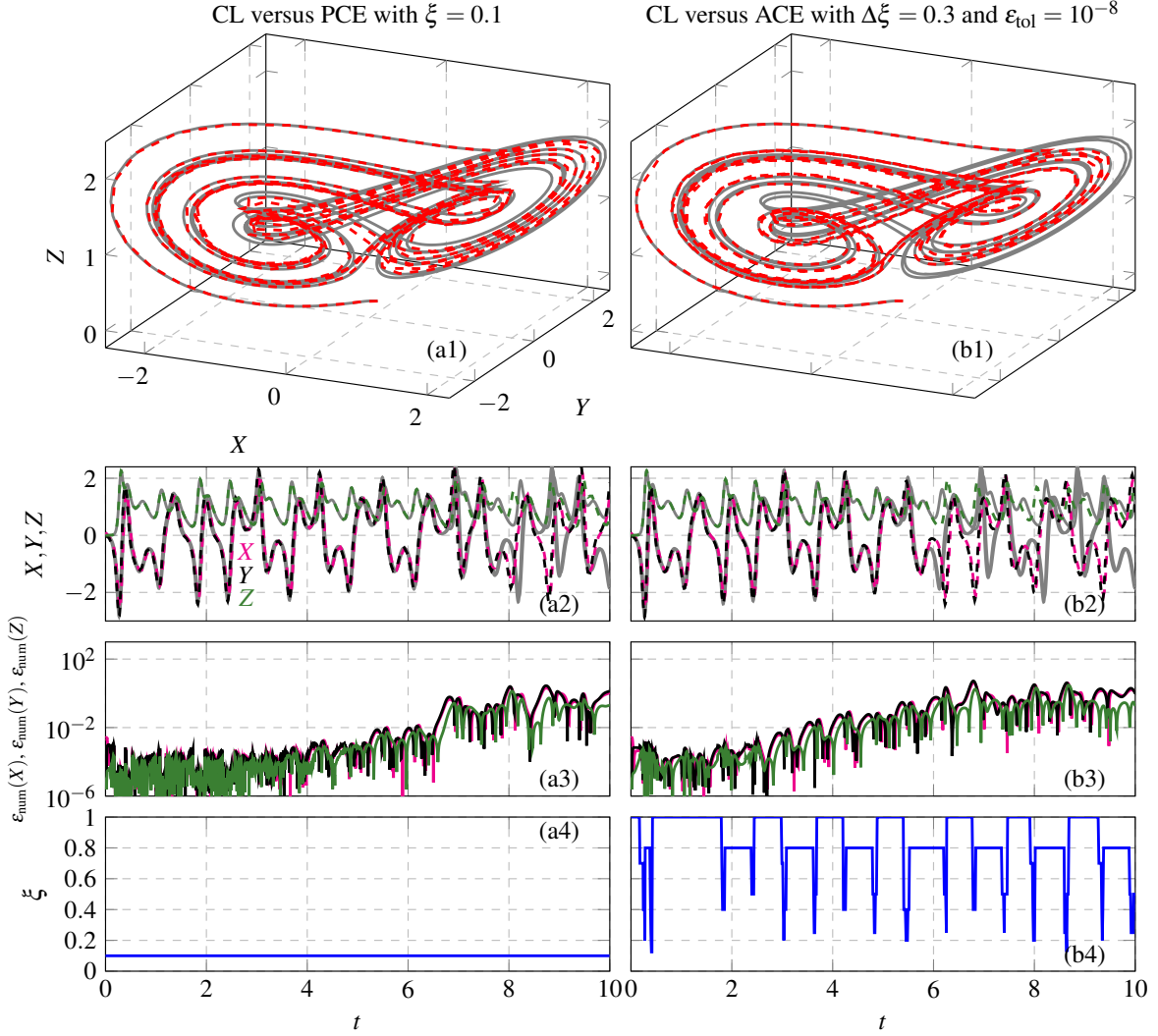


FIG. 10: Simulations of the Chen strange attractor (47) with $\sigma = 40$, $\rho = 28$, and $\beta = 6$. The initial conditions are $(X, Y, Z) = (0.1, 0.0, 0.0)$. The gray lines are CL simulations. The colored lines are PCE (a) or ACE (b) simulations. All Carleman simulations used $P = 6$. The ACE simulation used $\xi_{\text{init}} = \xi_{\text{max}} = 1.0$. The bottom row presents the temporal evolution of ξ .

$$d_t Z = XY - \beta Z. \quad (46c)$$

The variables X and Y are normalized to $\sqrt{\beta(2\rho - \sigma)}$, and the variable Z is normalized to $2\rho - \sigma$. After introducing the shifts ξ_x , ξ_y , and ξ_z , we rewrite the system as

$$d_t x = \sigma(\xi_y - \xi_x) - \sigma x + \sigma y, \quad (47a)$$

$$d_t y = [(\rho - \sigma)\xi_x + \rho\xi_y - (2\rho - \sigma)\xi_x\xi_z] + [(\rho - \sigma) - (2\rho - \sigma)\xi_z]x + \rho y - (2\rho - \sigma)\xi_x z - (2\rho - \sigma)xz, \quad (47b)$$

$$d_t z = (-\beta\xi_z + \beta\xi_x\xi_y) + \beta\xi_y x + \beta\xi_x y - \beta z + \beta xy. \quad (47c)$$

The results from modeling the Lorenz and Chen attractors are shown in Fig. 9 and Fig. 10, respectively. The standard embedding, not shown here, proves to be unstable for computing these systems. By contrast, the PCE and ACE

techniques succeed in bounding the modeling error in these chaotic systems.

We note that, due to the *butterfly effect*, which is expected from the theory of chaotic dynamical systems, for times longer than a few Lyapunov times, the numerical error using different numerical techniques becomes significant. Since these systems have nonzero Lyapunov exponents, even small differences in initial conditions or numerical algorithms will lead to a divergence in the simulated trajectories. The divergence manifests not only locally, but also globally, when at a given time instant two simulated trajectories are found on different loops of the attractor, i.e. near different fixed points. For instance, this can be observed in the ACE simulation with $\epsilon_{\text{tol}} = 10^{-8}$ of the Lorenz attractor shown in Fig. 9c2 where the X and Y signals modeled by ACE have an opposite sign compared to the CL signals starting at $t \approx 15$. This behavior is unavoidable, even when comparing stan-

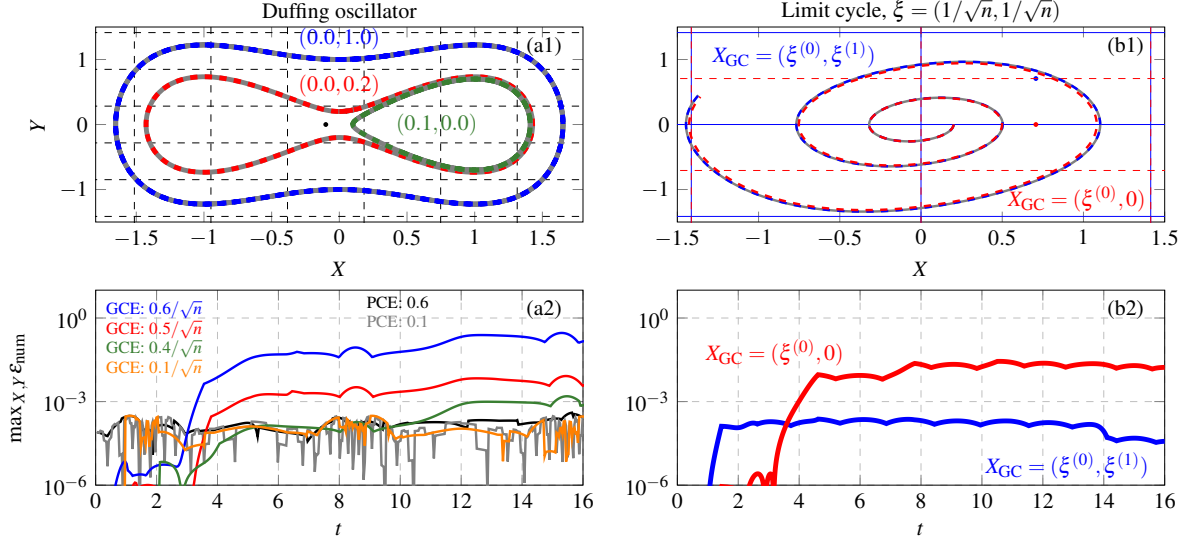


FIG. 11: (a1): Simulations of the Duffing oscillator with various initial conditions indicated by the colored text. The CL and GCE simulations are indicated by solid gray and dashed colored lines, respectively. All GCE simulations are performed in the same grid (black lines): $X_{GC} = (-0.1, 0.0)$ (black dot), $\xi^{(i)} = 0.4/\sqrt{n}$, with $i = 0, 1$. (a2): Numerical errors in the GCE and PCE simulations of the Duffing oscillator with the initial condition $(0.5, 0.5)$, the GCE grid center $X_{GC} = (-0.1, 0.0)$, and various sizes of the linearization regions (indicated by the colored text). (b1): Simulations of the Van der Pol limit cycle with the initial condition $(0.2, 0.0)$. The CL and GCE simulations are indicated by solid gray and dashed colored lines, respectively. The GCE simulations have the same tile sizes, $|\xi| = 1$, the grid centers are $X_{GC} = \xi$ (blue dot) and $X_{GC} = (\xi^{(0)}, 0)$ (red dot). The corresponding GCE grids are indicated by the solid blue and dashed red horizontal and vertical lines. (b2): Numerical errors in the GCE simulations of the limit cycle.

dard methods, because differences in floating-point precision alone can cause a chaotic dynamical system to follow a different trajectory.

For this reason, it may be reasonable to primarily use the PCE technique without attempting to adapt the convergence radius when modeling chaotic systems, as such adaptation might not significantly reduce the numerical error. However, ACE remains necessary in scenarios involving multiple fixed points, particularly when some are closely spaced and others are not. In these cases, ACE adjusts the linearization radius keeping it small near closely placed fixed points and increasing ξ in regions where fixed points are absent.

E. Testing the GCE technique

To test the GCE method, we start from two-dimensional systems such as the Van-der Pol limit cycle considered in Sec. III B and the Duffing oscillator. The Duffing oscillator,

$$d_t x = y, \quad (48a)$$

$$d_t y = x - x^3, \quad (48b)$$

has two centers at $(\pm 1, 0)$ and a saddle point at $(0, 0)$. After introducing the shifts ξ_x and ξ_y , one recasts the system as

$$d_t x = \xi_y + y, \quad (49a)$$

$$d_t y = (\xi_x - \xi_x^3) + (1 - 3\xi_x^2)x - 3\xi_x x^2 - x^3. \quad (49b)$$

The results from the GCE simulations of these 2D systems are shown in Fig. 11. As seen from Fig. 11a2, the GCE approach requires smaller linearization regions than the PCE method. For instance, the GCE simulation with $|\xi| = 0.6$ has numerical error $\mathcal{O}(1)$, while the PCE simulation with $\xi = 0.6$ has much smaller error around 10^{-2} . One possible reason for the stricter requirements on the chart size in the GCE method is that, during chart transitions, the PCE technique dynamically creates a new chart centered at the trajectory point just before it crosses the linearization boundary. According to the analysis 42 and 43, this procedure generates a much higher accuracy solution. In contrast, during a chart transition, the GCE technique moves from a point near the boundary of one tile to a point near the boundary in the neighboring tile, where the error is greatest. This can potentially lead to higher error accumulation in the GCE simulation, as the trajectory spends more time near the linearization horizon than in the PCE method.

One can also see from Figs. 11b1 and 11b2 that one can change the GCE accuracy by shifting the alignment of the trajectory and the center or edges of the linearization grid. In particular, when the grid center is shifted from $X_{GC} = (\xi^{(0)}, 0)$ (red grid) to $X_{GC} = (\xi^{(0)}, \xi^{(1)})$ (blue grid), the precision improves by several orders of magnitude. It is difficult to pinpoint the exact reason for this improvement. One possible explanation is that, in the case of the grid center $X_{GC} = (\xi^{(0)}, 0)$ (red grid), the trajectory of the limit cycle appears to spend more time near the top and bottom boundaries of the tiles $(0, 0)$ (the central tile) and $(-1, 0)$ (the tile to

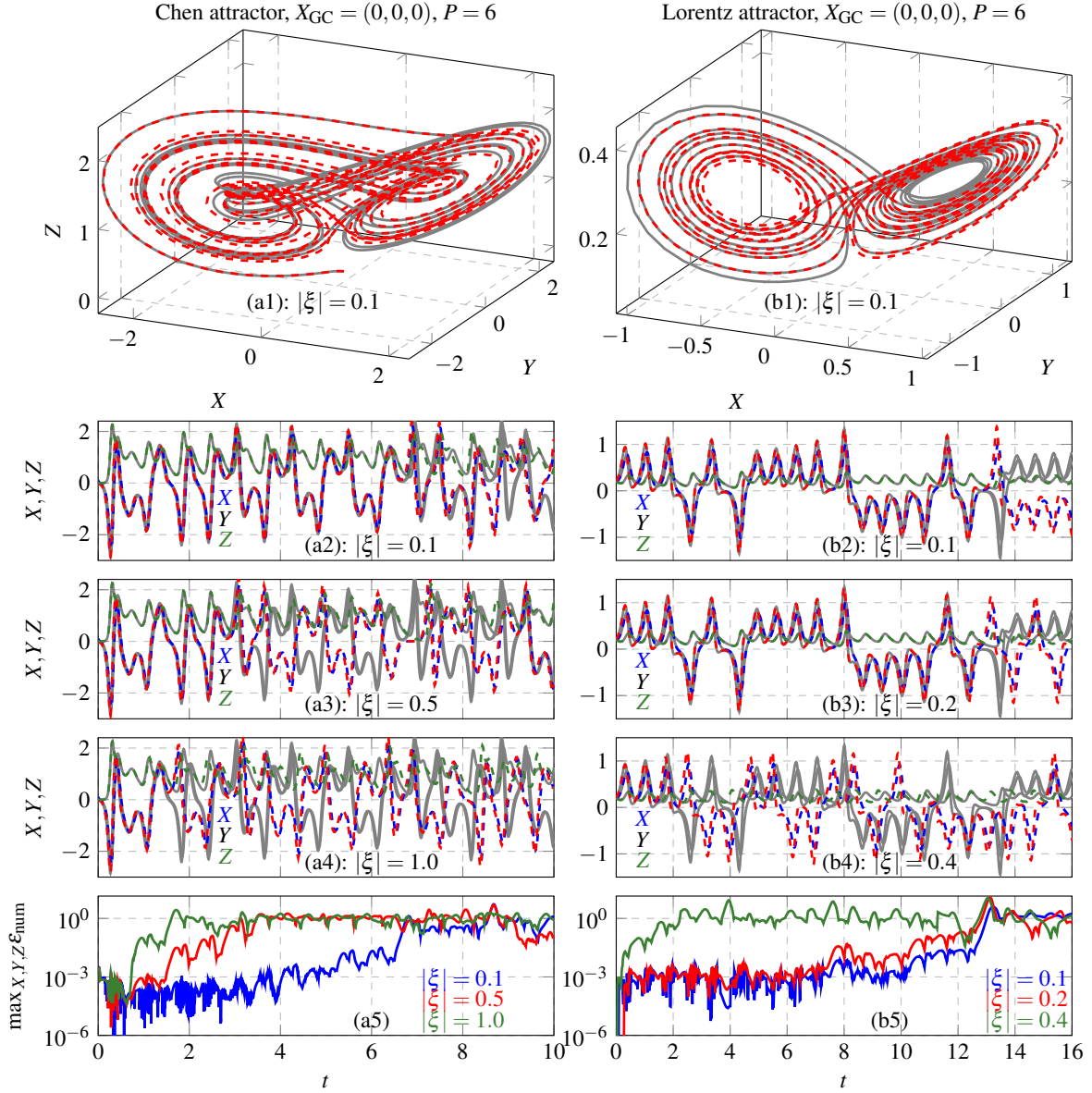


FIG. 12: (a1): Classical (gray line) and GCE (red line) simulations of the Chen strange attractor with $\sigma = 40$, $\rho = 28$, $\beta = 6$, and the initial condition $(0.1, 0.0, 0.0)$. (a2)-(a4): Comparison of the time evolution of individual variables (X, Y, Z) in classical simulations (gray lines) and GCE simulations for various tile sizes ξ (colored lines). (a5): Numerical error in the GCE simulations of the Chen attractor. (b1): Simulations of the Lorentz strange attractor with $\sigma = 10$, $\rho = 28$, $\beta = 8/3$, $c_z = 2c_x = 4$, and the initial condition $(0.2, 0.2, 0.2)$. (b2)-(b4): Comparison of the time evolution of individual variables in classical simulations (gray lines) and GCE simulations for various tile sizes ξ (colored lines). (b5): Numerical error in the GCE simulations of the Lorentz attractor. In all GCE simulations shown in plots (a) and (b), all components of ξ are set to $|\xi|/\sqrt{n}$, and the grid center is located at $(0, 0, 0)$.

the left). On the other hand, while the trajectory does spend some time near the corners of the tiles of the blue grid, it crosses the boundaries while moving rapidly in a direction that is nearly perpendicular to the grid rather than spending a long time moving parallel to the edges where the error is large.

We also test the GCE method on simulations of the Chen and Lorentz attractors described in Sec. III D. The results of the GCE simulations are shown in Fig. 12. The error of the

GCE simulations of the Chen attractor remains bounded for the maximum allowable tile size equal to $|\xi| = 1.0$, with all components of the vector ξ set equal. Comparing Fig. 10a3 with Fig. 12a5 shows that the PCE and GCE simulations of the Chen attractor exhibit similar numerical errors for the same chart size, $|\xi| = 0.1$. Similarly, a comparison between Fig 9a3 and Fig. 12b5 reveals that the PCE and GCE simulations of the Lorentz attractor yield comparable numerical errors at the same chart size $|\xi| = 0.1$.

The Lorentz attractor appears to be a more challenging nonlinear system to model using Carleman linearization, as it requires smaller tiles than those used in the GCE simulations of the Chen attractor. Although the numerical error remains bounded even for $|\xi| = 1.0$, the trajectories of the Lorentz attractor simulated with such a GCE grid differ drastically from the classical trajectories for the chosen initial conditions.

IV. CONCLUSION

We have demonstrated that the Carleman embedding can be globalized to significantly enhance the accuracy and bound the maximum error when simulating nonlinear systems with multiple fixed points, even in cases with chaotic behavior. By partitioning the state space into multiple linearization charts, we successfully stabilize the Carleman method for many cases that are otherwise intractable for the standard Carleman linearization.

To avoid large errors in cases where multiple nearby fixed points appear within the same linearization chart, we designed an adaptive technique that dynamically adjusts the chart radius ξ by analyzing the nonlinear terms in the Carleman embedding and by comparing two embeddings with different convergence radii. This technique either reduces the convergence radius to minimize numerical error or enlarges the embedding charts to accelerate simulation speed. This adaptive adjustment enables the identification of more optimal linearization chart sizes for different segments of the simulated trajectory. However, numerical tests indicate that, in the case of chaotic systems, the non-adaptive technique with a small fixed linearization radius may perform faster with a comparable precision, particularly when prior knowledge of the minimal required distance between fixed points in the simulated system is available.

The adaptive methods do not require prior knowledge of the locations of the fixed points, but they do require careful detection of the times when it is necessary to switch between linearization charts. The accuracy of the method depends not only on the truncation of the maximum monomial order, but also on the imposed convergence radius of the linearization regions in the non-adaptive version and on the tolerance specified in the adaptive version. Through various numerical simulations, we have demonstrated that the numerical error in this globalized Carleman approach remains bounded even for chaotic dynamical systems, including strange attractors of various types.

The standard Carleman embedding is well suited for methods that exploit linear systems well, such as quantum computing and tensor networks, as it only requires solving a linear initial-value problem (7). In contrast, the globalized Carleman method requires careful detection of chart boundary crossings. The adaptive version involves an even greater number of nonlinear operations, making it considerably less suitable for the development of a quantum implementation. Therefore, we consider the non-adaptive piecewise technique as the primary candidate for constructing a globalized Carleman quantum algorithm for simulating nonlinear dynamics.

However, the question of whether an efficient quantum algorithm based on the proposed piecewise technique can be realized remains open. The challenge lies in the fact that this method requires frequent evaluation of either the vector norm in the dynamic approach (PCE) or the vector components in the static piecewise version (GCE), in order to detect when the simulated trajectory crosses the boundary of a linearization chart. This operation poses challenges for quantum computers and is most likely implementable only through measurements.

A possible approach for circumventing this requirement is to implement the globalized Carleman algorithm as a hybrid classical-quantum algorithm where the classical computer performs the nonlinear work. The strategy is to solve the system (7) on a quantum computer for a fixed time interval, then measure the local variables x and classically analyze whether the system remains within the initial linearization chart. If this is the case, a new chart is constructed at the measured point, and quantum Hamiltonian simulation is performed within this newly created chart for the next fixed time interval, after which the process repeats. If the measured variables fall outside the chart, the results are discarded, and the system (7) is solved within the same chart over a shorter time interval.

For a hybrid algorithm, the primary limiting factor for achieving speedup will be the data exchange between the classical and quantum processors. However, the only data that needs to be exchanged during each chart shift are the displacements x_ξ or the components of the local n -dimensional vector x . This is always much smaller than size of the full Carleman vector $N = C_n^{n+P}$. For the regime $P \gg n$ required for low-dimensional ODEs, where $N \sim P^n$, it is logarithmically smaller in dimension, n . For the regime $n \gg P$ required for high-dimensional ODEs, such as the discretization of PDEs, where $N \sim n^P$, it is logarithmically smaller in polynomial degree, P . Thus, the relatively small amount of data exchange required is potentially compatible with the constraints of an efficient hybrid quantum-classical algorithm.

On the other hand, the fact that the amount of measured data scales linearly with n implies that the number of variables to be measured increases linearly with the size of the simulated nonlinear problem. Thus, for instance, when simulating a set of PDEs, such as the Vlasov-Maxwell system with (N_x, N_v) variables in phase space, one would need to measure $N_x \times N_v$ variables. Considering this cost, the necessity of using a globalized Carleman embedding is debatable, especially given that the KvN method already provides a global embedding of nonlinear problems, for which a highly efficient quantum algorithm has already been provided²⁶. Yet, the power of the global Carleman method motivates the exploration of finite element and spectral element methods for the KvN approach to generating a linear embedding of the Koopman and Perron-Frobenius evolution operators.

ACKNOWLEDGMENTS

We would like to thank Matthew Kvalheim and Hong Qin for discussions regarding the minimal number of charts

needed to linearize a vector field on a manifold. We would also like to thank Matteo Lostaglio and Yiğit Subaşı for pointing out that the minimal size of the Carleman vector is equal to the number of distinct homogeneous polynomials up to a given order. This work, LLNL-JRNL-2010809, was supported by the U.S. Department of Energy (DOE) Office of Fusion Energy Sciences “Quantum Leap for Fusion Energy Sciences” Project No. FWP-SCW1680 at Lawrence Livermore National Laboratory (LLNL). Work was performed under the auspices of the U.S. DOE under LLNL Contract DE-AC52-07NA27344.

Appendix A: Applying the Standard Carleman Embedding to 1D Flows

Consider an autonomous 1D vector field $V(x)$ that can be expressed as a p -th order polynomial in x . The Carleman procedure constructs the coordinate transformation $x(t, x_0)$ as a power series in x_0 that can be numerically integrated in time. However, the Carleman procedure must fail whenever this relationship is beyond the radius of convergence of the power series. This is determined by the radius in the complex plane where the coordinate transformation is no longer analytic in x_0 .

For simplicity, first consider the case where $V(x)$ has simple zeros and $p \geq 1$. The solution to the differential equation

$$dx/dt = V(x) \quad (\text{A1})$$

can be determined by dividing by the velocity and integrating to find the time as a function of x ,

$$t - t_0 = \int_{x_0}^{x_t} dx/V(x). \quad (\text{A2})$$

Using the partial fraction decomposition of $1/V(x)$ allows one to explicitly write the solution as

$$t - t_0 = \int_{x_0}^{x_t} \sum_{j=1}^n \tau_j dx/(x - x_j) = \sum_{j=1}^n \tau_j \log \left| \frac{x_t - x_j}{x_0 - x_j} \right|, \quad (\text{A3})$$

where the fixed points x_j for $j = 1, 2, \dots, n$ are ordered by their distance to the initial condition, x_0 .

For a single fixed point, the solution is simply

$$x(t) = x_1 + (x_0 - x_1)e^{(t-t_0)/\tau_1}. \quad (\text{A4})$$

The Carleman method constructs an approximate Taylor series of Eq. A4, and, hence, the Carleman procedure can converge over the entire orbit, as long as the Carleman radius ξ is larger than $|x_0 - x_1|$. This is because, if $\tau_1 < 0$ (> 0) the future (past) trajectory is guaranteed to remain within the Carleman disk. However, as soon as there is more than one fixed point, the radius of convergence of Eq. A3 will be limited by the second closest fixed point to the origin, x_2 .

These arguments can be generalized to the case where $V(x)$ is a rational function with a polynomial numerator of order n and a polynomial denominator of order m . In this

case, the partial fraction expansion of $1/V(x)$ yields the result

$$\begin{aligned} t - t_0 &= \int_{x_0}^x dx/V(x) = \int_{x_0}^x dx \sum_j \tau_j (x - x_j)^{-p_j} + c_j x^j \\ &= \sum_j \tau_j \frac{((x - x_j)^{1-p_j} - (x_0 - x_j)^{1-p_j})}{(1-p_j)} + c_j \frac{x^{j+1} - x_0^{j+1}}{(j+1)}, \end{aligned} \quad (\text{A5})$$

where $n = \sum_j p_j$. Clearly, at each x_j , the series has poles where $p_j \geq 2$ and branch cuts where $p_j = 1$ or is not an integer. For a single fixed point, the solution is constructed correctly. However, when there are two fixed points, the radius of convergence is limited to the second nearest fixed point to the origin, x_2 .

First, consider the case where x_1 is stable and x_2 is unstable. For an initial condition of the form $x_1 < x_0 < x_2$, the trajectory is attracted to x_1 . Convergence is worst near x_0 , but improves as x travels towards x_1 . For $x_0 < x_1$, the trajectory is still attracted to x_1 , but convergence is worst where x is furthest from the origin. However, for $x_0 > x_2$, convergence cannot be achieved as the trajectory flies off towards $x \rightarrow \infty$ and, in fact, the solution blows up as $x \rightarrow 1$.

Second, consider the case where x_1 is unstable and x_2 is stable. Now, for an initial condition of the form $x_1 < x_0 < x_2$, the trajectory is repelled from x_1 and attracted to x_2 . However, if the number of polynomials is fixed, then the method always fails once the trajectory gets close enough to x_2 . If $x_0 < 0$, then the trajectory flies off towards $x \rightarrow -\infty$, but the evolution in time abruptly accumulates large errors once $x < -x_2$. After this point, convergence can no longer be achieved and the solution blows up as $x \rightarrow -1$. Clearly, convergence also cannot be achieved in the case where $x_0 > x_2$ and the solution blows up for $x \rightarrow 1$.

We have performed a series of tests to verify all of these cases. The predictions above correctly describe our numerical experiments. For example, numerical evidence in Sec. IIB shown in Fig. 4(a) (black dashed lines) demonstrates that convergence is difficult to achieve even for the region $x_1 < x < x_2$. In this case, convergence can be achieved if more polynomials are retained; we found $P = 20$ and above was sufficient.

Even for a single fixed point of order $p_j > 1$, the standard Carleman method fails unless one modifies the formulation. In this case, one can linearize the region near the fixed point using the non-analytic change of variables

$$(p_j - 1) \log |z| = \Delta x / |\Delta x|^{p_j} \quad (\text{A6})$$

$$|z|^{p_j-1} = \exp(\Delta x / |\Delta x|^{p_j}), \quad (\text{A7})$$

where, here, $\Delta x = x - x_j$. In this formula, we used the choice of sign conventions to handle the region below, $x < x_j$, and above, $x_j < x$, the fixed point to correspond to $z < 1$ and $1 < z$, respectively.

This formula also holds when $p_j < 1$ and even for a singular fixed point where $p_j < 0$. In this case, the linearizing transformation for $p_j < 1$ can be rewritten as

$$(1 - p_j) \log |z| = \Delta x |\Delta x|^{-p_j}, \quad (\text{A8})$$

$$|z|^{1-p_j} = \exp(\Delta x |\Delta x|^{-p_j}). \quad (\text{A9})$$

Hence, it is not as singular as the case of $p_j > 1$, and, can even be chosen to be analytic when p_j is a negative even integer.

If the partial fraction expansion of $V(x)$ includes a polynomial of order $p_\infty \geq 1$, then the singular point at infinity can be handled in a similar fashion. The change of variables $y = 1/x$ repositions the singular point at $x = \pm\infty$ to the point $y = 0$ and changes the effective order of the polynomial to $\tilde{y} \propto y^{2-p_\infty}$. Hence, this can be handled as a singular point of order $2 - p_\infty$, which leads to the linearizing transformation

$$(p_\infty + 1) \log |z| = -y |y|^{p_\infty} = -1/x |x|^{p_\infty}, \quad (\text{A10})$$

$$|z|^{p_\infty+1} = \exp(-y |y|^{p_\infty}) \quad (\text{A11})$$

$$= \exp\left(-1/x |x|^{p_\infty}\right). \quad (\text{A12})$$

In this case, our choice of sign conventions handles the regions $-\infty < x < 0$ and $0 < x < \infty$ with the regions $z < 1$ and $z > 1$, respectively.

¹T. Carleman, Application de la théorie des équations intégrales linéaires aux systèmes d'équations différentielles non linéaires, *Acta Mathematica* 59 (1932) 63–87. doi:10.1007/BF02546499. URL <https://doi.org/10.1007/BF02546499>

²W. Steeb, Embedding of nonlinear finite dimensional systems in linear infinite dimensional systems and bose operators, *Hadronic J.* 6 (1983) 68–76.

³K. Kowalski, Hilbert space description of classical dynamical systems I, *Physica A: Statistical Mechanics and its Applications* 145 (3) (1987) 408–424. doi:https://doi.org/10.1016/0378-4371(87)90003-3. URL <https://www.sciencedirect.com/science/article/pii/0378437187900033>

⁴A. Engel, G. Smith, S. E. Parker, Linear embedding of nonlinear dynamical systems and prospects for efficient quantum algorithms, *Physics of Plasmas* 28 (6) (2021) 062305. arXiv:https://pubs.aip.org/aip/pop/article-pdf/doi/10.1063/5.0040313/13313306/062305_1_online.pdf, doi:10.1063/5.0040313. URL <https://doi.org/10.1063/5.0040313>

⁵Y. T. Lin, R. B. Lowrie, D. Aslangil, Y. Subaşı, A. T. Sornborger, Koopman–von Neumann mechanics and the Koopman representation: A perspective on solving nonlinear dynamical systems with quantum computers (2022). arXiv:2202.02188.

⁶I. Joseph, Y. Shi, M. D. Porter, A. R. Castelli, V. I. Geyko, F. R. Graziani, S. B. Libby, J. L. DuBois, Quantum computing for fusion energy science applications, *Physics of Plasmas* 30 (1) (Jan. 2023). doi:10.1063/5.0123765. URL <http://dx.doi.org/10.1063/5.0123765>

⁷I. Mezić, Spectral properties of dynamical systems, model reduction and decompositions, *Nonlinear Dynamics* 41 (1) (2005) 309–325. doi:10.1007/s11071-005-2824-x. URL <https://doi.org/10.1007/s11071-005-2824-x>

⁸S. L. Brunton, M. Budišić, E. Kaiser, J. N. Kutz, Modern koopman theory for dynamical systems, *SIAM Review* 64 (2) (2022) 229–340. arXiv:https://doi.org/10.1137/21M1401243, doi:10.1137/21M1401243. URL <https://doi.org/10.1137/21M1401243>

⁹M. J. Colbrook, I. Mezić, A. Stepanenko, Limits and powers of koopman learning, arXiv preprint arXiv:2407.06312 (2024).

¹⁰M. Korda, I. Mezić, Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control, *Automatica* 93 (2018) 149–160. doi:https://doi.org/10.1016/j.automatica.2018.03.046. URL <https://www.sciencedirect.com/science/article/pii/S000510981830133X>

¹¹M. Korda, I. Mezić, Optimal construction of Koopman eigenfunctions for prediction and control (2020). arXiv:1810.08733. URL <https://arxiv.org/abs/1810.08733>

¹²B. O. Koopman, Hamiltonian systems and transformations in Hilbert space, *Proceedings of the National Academy of Sciences of the United States of America* 17 (5) (1931) 315–318. doi:10.1073/pnas.17.5.315. URL <https://www.pnas.org/doi/10.1073/pnas.17.5.315>

¹³B. O. Koopman, J. v. Neumann, Dynamical systems of continuous spectra, *Ann. Math.* 18 (1932) 255.

¹⁴I. Joseph, Koopman–von Neumann approach to quantum simulation of nonlinear classical dynamics, *Phys. Rev. Res.* 2 (2020) 043102. doi:10.1103/PhysRevResearch.2.043102. URL <https://link.aps.org/doi/10.1103/PhysRevResearch.2.043102>

¹⁵J.-P. Liu, H. Kolden, H. K. Krovi, N. F. Loureiro, K. Trivisa, A. M. Childs, Efficient quantum algorithm for dissipative nonlinear differential equations, *Proceedings of the National Academy of Sciences* 118 (35) (Aug. 2021). doi:10.1073/pnas.2026805118. URL <http://dx.doi.org/10.1073/pnas.2026805118>

¹⁶H. Krovi, Improved quantum algorithms for linear and nonlinear differential equations, *Quantum* 7 (2023) 913. doi:10.22331/q-2023-02-02-913. URL <http://dx.doi.org/10.22331/q-2023-02-02-913>

¹⁷T. Vaszary, A. Datta, T. Goffrey, B. Appelbe, Solving the nonlinear Vlasov equation on a quantum computer (2025). arXiv:2411.19310. URL <https://arxiv.org/abs/2411.19310>

¹⁸A. Ambainis, Variable time amplitude amplification and quantum algorithms for linear algebra problems, in: C. Dürr, T. Wilke (Eds.), 29th International Symposium on Theoretical Aspects of Computer Science (STACS 2012), Vol. 14 of Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2012, pp. 636–647. doi:10.4230/LIPIcs.STACS.2012.636. URL <http://drops.dagstuhl.de/opus/volltexte/2012/3426>

¹⁹L. Lin, Y. Tong, Optimal polynomial based quantum eigenstate filtering with application to solving quantum linear systems, *Quantum* 4 (2020) 361. doi:10.22331/q-2020-11-11-361. URL <https://doi.org/10.22331/q-2020-11-11-361>

²⁰G. H. Low, Y. Su, Quantum linear system algorithm with optimal queries to initial state preparation (2024). arXiv:2410.18178. URL <https://arxiv.org/abs/2410.18178>

²¹A. M. Dalzell, A shortcut to an optimal quantum linear system solver (2024). arXiv:2406.12086. URL <https://arxiv.org/abs/2406.12086>

²²S. Jin, N. Liu, Y. Yu, Quantum simulation of partial differential equations via schrödingerization, *Phys. Rev. Lett.* 133 (2024) 230602. doi:10.1103/PhysRevLett.133.230602. URL <https://link.aps.org/doi/10.1103/PhysRevLett.133.230602>

²³D. An, J.-P. Liu, L. Lin, Linear combination of hamiltonian simulation for nonunitary dynamics with optimal state preparation cost, *Phys. Rev. Lett.* 131 (2023) 150603. doi:10.1103/PhysRevLett.131.150603. URL <https://link.aps.org/doi/10.1103/PhysRevLett.131.150603>

²⁴D. An, A. M. Childs, L. Lin, Quantum algorithm for linear non-unitary dynamics with near-optimal dependence on all parameters (2023). arXiv:2312.03916.

²⁵I. Joseph, Semiclassical theory and the Koopman–van Hove equation, *Journal of Physics A: Mathematical and Theoretical* 56 (48) (2023) 484001. doi:10.1088/1751-8121/ad0533. URL <http://dx.doi.org/10.1088/1751-8121/ad0533>

²⁶I. Novikau, I. Joseph, Quantum algorithm for the advection-diffusion equation and the Koopman–von Neumann approach to nonlinear dynamical systems, *Computer Physics Communications* 309 (2025) 109498. doi:https://doi.org/10.1016/j.cpc.2025.109498. URL <https://www.sciencedirect.com/science/article/pii/S0010465525000013>

²⁷I. Novikau, I. Joseph, An efficient explicit implementation of a near-optimal quantum algorithm for simulating linear dissipative differential equations (2025). arXiv:2501.11146.

- URL <https://arxiv.org/abs/2501.11146>
- ²⁸S. Succi, W. Itani, K. Sreenivasan, R. Steijl, Quantum computing for fluids: Where do we stand?, *Europhysics Letters* 144 (1) (2023) 10001. doi:10.1209/0295-5075/acfdc7.
URL <https://dx.doi.org/10.1209/0295-5075/acfdc7>
- ²⁹C. Sanavio, R. Scatamacchia, C. de Falco, S. Succi, Three carleman routes to the quantum simulation of classical fluids, *Physics of Fluids* 36 (5) (2024) 057143. arXiv:https://pubs.aip.org/aip/pof/article-pdf/doi/10.1063/5.0204955/19963101/057143_1_5.0204955.pdf, doi:10.1063/5.0204955.
URL <https://doi.org/10.1063/5.0204955>
- ³⁰C. Sanavio, E. Mauri, S. Succi, Explicit quantum circuit for simulating the advection–diffusion–reaction dynamics, *IEEE Transactions on Quantum Engineering* 6 (2025) 1–12. doi:10.1109/tqe.2025.3544839.
URL <http://dx.doi.org/10.1109/TQE.2025.3544839>
- ³¹J. Gonzalez-Conde, D. Lewis, S. S. Bharadwaj, M. Sanz, Quantum carleman linearization efficiency in nonlinear fluid dynamics, *Phys. Rev. Res.* 7 (2025) 023254. doi:10.1103/PhysRevResearch.7.023254.
URL <https://link.aps.org/doi/10.1103/PhysRevResearch.7.023254>
- ³²X. Li, X. Yin, N. Wiebe, J. Chun, G. K. Schenter, M. S. Cheung, J. Mülmenstädt, Potential quantum advantage for simulation of fluid dynamics, *Phys. Rev. Res.* 7 (2025) 013036. doi:10.1103/PhysRevResearch.7.013036.
URL <https://link.aps.org/doi/10.1103/PhysRevResearch.7.013036>
- ³³T. Akiba, Y. Morii, K. Maruta, Carleman linearization approach for chemical kinetics integration toward quantum computation, *Scientific Reports* 13 (1) (2023) 3935. doi:10.1038/s41598-023-31009-9.
URL <https://doi.org/10.1038/s41598-023-31009-9>
- ³⁴T. Vaszary, Carleman linearization of partial differential equations (2024). arXiv:2412.00014.
URL <https://arxiv.org/abs/2412.00014>
- ³⁵J.-P. Liu, D. An, D. Fang, J. Wang, G. H. Low, S. Jordan, Efficient quantum algorithm for nonlinear reaction–diffusion equations and energy estimation, *Communications in Mathematical Physics* 404 (2) (2023) 963–1020. doi:10.1007/s00220-023-04857-9.
URL <https://doi.org/10.1007/s00220-023-04857-9>
- ³⁶D. Jennings, K. Korzekwa, M. Lostaglio, A. T. Sornborger, Y. Subasi, G. Wang, Quantum algorithms for general nonlinear dynamics based on the carleman embedding (2025). arXiv:2509.07155.
URL <https://arxiv.org/abs/2509.07155>
- ³⁷N. Motee, Q. Sun, Carleman-fourier linearization of nonlinear real dynamical systems with quasi-periodic fields (2025). arXiv:2503.01498.
URL <https://arxiv.org/abs/2503.01498>
- ³⁸P. Arathoon, M. D. Kvalheim, Koopman embedding and super-linearization counterexamples with isolated equilibria (2023). arXiv:2306.15126.
URL <https://arxiv.org/abs/2306.15126>
- ³⁹M.-A. Belabbas, X. Chen, A sufficient condition for the super-linearization of polynomial systems, *Systems & Control Letters* 179 (2023) 105588. doi:<https://doi.org/10.1016/j.sysconle.2023.105588>.
URL <https://www.sciencedirect.com/science/article/pii/S0167691123001354>
- ⁴⁰T. Breunung, F. Kogelbauer, Learning global linear representations of nonlinear dynamics (2024). arXiv:2408.03437.
URL <https://arxiv.org/abs/2408.03437>
- ⁴¹M. Forets, A. Pouly, Explicit error bounds for carleman linearization (2017). arXiv:1711.02552.
URL <https://arxiv.org/abs/1711.02552>
- ⁴²H. Weber, W. Mathis, Adapting the range of validity for the carleman linearization, *Advances in Radio Science* 14 (2016) 51–54. doi:10.5194/ars-14-51-2016.
URL <https://ars.copernicus.org/articles/14/51/2016/>
- ⁴³J. C. Muñoz Sánchez, S. F. Elena, J.-A. Oteo, Carleman approximants for non-linear differential systems, *Physica Scripta* 100 (6) (2025) 065231. doi:10.1088/1402-4896/add65a.
URL <https://dx.doi.org/10.1088/1402-4896/add65a>
- ⁴⁴A. Mauroy, Y. Susuki, I. Mezic, Koopman operator in systems and control, Vol. 7, Springer, 2020.
- ⁴⁵D. Shi, X. Yang, Koopman spectral linearization vs. Carleman linearization: A computational comparison study, *Mathematics* 12 (14) (2024). doi:10.3390/math12142156.
URL <https://www.mdpi.com/2227-7390/12/14/2156>
- ⁴⁶M. D. Kvalheim, P. Arathoon, Linearizability of flows by embeddings (2024). arXiv:2305.18288.
URL <https://arxiv.org/abs/2305.18288>
- ⁴⁷Z. Liu, N. Ozay, E. D. Sontag, On the non-existence of immersions for systems with multiple omega-limit sets, *IFAC-PapersOnLine* 56 (2) (2023) 60–64, 22nd IFAC World Congress. doi:<https://doi.org/10.1016/j.ifacol.2023.10.1408>.
URL <https://www.sciencedirect.com/science/article/pii/S2405896323018165>
- ⁴⁸Z. Liu, N. Ozay, E. D. Sontag, Properties of immersions for systems with multiple limit sets with implications to learning koopman embeddings, *Automatica* 176 (2025) 112226. doi:<https://doi.org/10.1016/j.automatica.2025.112226>.
URL <https://www.sciencedirect.com/science/article/pii/S0005109825001189>
- ⁴⁹P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, *Journal of Fluid Mechanics* 656 (2010) 5–28. doi:10.1017/S0022112010001217.
- ⁵⁰M. Budišić, R. Mohr, I. Mezić, Applied Koopmanism, *Chaos: An Interdisciplinary Journal of Nonlinear Science* 22 (4) (2012) 047510. arXiv:https://pubs.aip.org/aip/cha/article-pdf/doi/10.1063/1.4772195/13471578/047510_1_online.pdf, doi:10.1063/1.4772195.
URL <https://doi.org/10.1063/1.4772195>
- ⁵¹S. L. Brunton, J. L. Proctor, J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proceedings of the national academy of sciences* 113 (15) (2016) 3932–3937. doi:10.1073/pnas.1517384113.
URL <https://doi.org/10.1073/pnas.1517384113>
- ⁵²S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, J. N. Kutz, Chaos as an intermittently forced linear system, *Nature Communications* 8 (1) (May 2017). doi:10.1038/s41467-017-00030-8.
URL <http://dx.doi.org/10.1038/s41467-017-00030-8>
- ⁵³S. L. Brunton, B. R. Noack, P. Koumoutsakos, Machine learning for fluid mechanics, *Annual Review of Fluid Mechanics* 52 (Volume 52, 2020) (2020) 477–508. doi:<https://doi.org/10.1146/annurev-fluid-010719-060214>.
URL <https://www.annualreviews.org/content/journals/10.1146/annurev-fluid-010719-060214>
- ⁵⁴R. Taylor, J. N. Kutz, K. Morgan, B. A. Nelson, Dynamic mode decomposition for plasma diagnostics and validation, *Review of Scientific Instruments* 89 (5) (2018) 053501. arXiv:https://pubs.aip.org/aip/rsi/article-pdf/doi/10.1063/1.5027419/15631032/053501_1_online.pdf, doi:10.1063/1.5027419.
URL <https://doi.org/10.1063/1.5027419>
- ⁵⁵M. Sasaki, Y. Kawachi, R. O. Dendy, H. Arakawa, N. Kasuya, F. Kin, K. Yamasaki, S. Inagaki, Using dynamical mode decomposition to extract the limit cycle dynamics of modulated turbulence in a plasma simulation, *Plasma Physics and Controlled Fusion* 61 (11) (2019) 112001. doi:10.1088/1361-6587/ab471b.
URL <https://dx.doi.org/10.1088/1361-6587/ab471b>
- ⁵⁶I. Nayak, M. Kumar, F. L. Teixeira, Detection and prediction of equilibrium states in kinetic plasma simulations via mode tracking using reduced-order dynamic mode decomposition, *Journal of Computational Physics* 447 (2021) 110671. doi:<https://doi.org/10.1016/j.jcp.2021.110671>.
URL <https://www.sciencedirect.com/science/article/pii/S0021999121005660>
- ⁵⁷A. Kusaba, T. Kuboyama, K. Shin, M. Sasaki, S. Inagaki, A new combination of hankel and sparsity-promoting dynamic mode decompositions and its application to the prediction of plasma turbulence, *Japanese Journal of Applied Physics* 61 (SA) (2021) SA1011. doi:10.35848/1347-4065/ac1c3c.
URL <https://dx.doi.org/10.35848/1347-4065/ac1c3c>
- ⁵⁸L. E. di Grazia, M. Mattei, A. Mele, A. Pironti, A data-driven vertical stabilization system for the iter tokamak based on dynamic mode de-

- composition, *Journal of the Franklin Institute* 361 (2) (2024) 816–833. doi:<https://doi.org/10.1016/j.jfranklin.2023.12.027>. URL <https://www.sciencedirect.com/science/article/pii/S0016003223007974>
- ⁵⁹A. A. Kaptanoglu, K. D. Morgan, C. J. Hansen, S. L. Brunton, Characterizing magnetized plasmas with dynamic mode decomposition, *Physics of Plasmas* 27 (3) (2020) 032108. arXiv:https://pubs.aip.org/aip/pop/article-pdf/doi/10.1063/1.5138932/13806412/032108_1_online.pdf, doi:10.1063/1.5138932. URL <https://doi.org/10.1063/1.5138932>
- ⁶⁰F. Faraji, M. Reza, A. Knoll, J. N. Kutz, Dynamic mode decomposition for data-driven analysis and reduced-order modeling of $e \times b$ plasmas: I. extraction of spatiotemporally coherent patterns, *Journal of Physics D: Applied Physics* 57 (6) (2023) 065201. doi:10.1088/1361-6463/ad0910. URL <https://dx.doi.org/10.1088/1361-6463/ad0910>
- ⁶¹F. Faraji, M. Reza, A. Knoll, J. N. Kutz, Dynamic mode decomposition for data-driven analysis and reduced-order modeling of $e \times b$ plasmas: II. dynamics forecasting, *Journal of Physics D: Applied Physics* 57 (6) (2023) 065202. doi:10.1088/1361-6463/ad0911. URL <https://dx.doi.org/10.1088/1361-6463/ad0911>
- ⁶²S. De Pascuale, D. L. Green, J. D. Lore, Data-driven linear time advance operators for the acceleration of plasma physics simulation, *Physics of Plasmas* 29 (11) (2022) 113903. arXiv:https://pubs.aip.org/aip/pop/article-pdf/doi/10.1063/5.0110393/16778809/113903_1_online.pdf, doi:10.1063/5.0110393. URL <https://doi.org/10.1063/5.0110393>
- ⁶³I. Nayak, F. L. Teixeira, D.-Y. Na, M. Kumar, Y. A. Omelchenko, Accelerating particle-in-cell kinetic plasma simulations via reduced-order modeling of space-charge dynamics using dynamic mode decomposition, *Phys. Rev. E* 109 (2024) 065307. doi:10.1103/PhysRevE.109.065307. URL <https://link.aps.org/doi/10.1103/PhysRevE.109.065307>
- ⁶⁴M. Lo Verso, S. Riva, C. Intorini, E. Cervi, L. Barucca, M. Caramello, M. Di Prinzio, F. Giacobbo, L. Savoldi, A. Cammi, Enhancing computational efficiency in nuclear fusion through reduced order modelling: Applications in magnetohydrodynamics, *Fusion Engineering and Design* 216 (2025) 115080. doi:<https://doi.org/10.1016/j.fusengdes.2025.115080>. URL <https://www.sciencedirect.com/science/article/pii/S0920379625002777>
- ⁶⁵A. V. Dudkovskaia, D. Li, J. Candy, E. A. Belli, C. Yang, Dynamic mode decomposition for gyrokinetic eigenmode analysis, *Plasma Physics and Controlled Fusion* 67 (6) (2025) 065033. doi:10.1088/1361-6587/adb73. URL <https://dx.doi.org/10.1088/1361-6587/adb73>
- ⁶⁶P. Ji, J. Ye, Y. Mu, W. Lin, Y. Tian, C. Hens, M. Perc, Y. Tang, J. Sun, J. Kurths, Signal propagation in complex networks, *Physics Reports* 1017 (2023) 1–96, signal propagation in complex networks. doi:<https://doi.org/10.1016/j.physrep.2023.03.005>. URL <https://www.sciencedirect.com/science/article/pii/S0370157323001321>
- ⁶⁷A. T. Taylor, T. A. Berrueta, T. D. Murphey, Active learning in robotics: A review of control principles, *Mechatronics* 77 (2021) 102576. doi:<https://doi.org/10.1016/j.mechatronics.2021.102576>. URL <https://www.sciencedirect.com/science/article/pii/S0957415821000659>
- ⁶⁸O. Azencot, N. B. Erichson, V. Lin, M. W. Mahoney, Forecasting sequential data using consistent Koopman autoencoders (2020). arXiv:2003.02236. URL <https://arxiv.org/abs/2003.02236>
- ⁶⁹K. Fukami, T. Nakamura, K. Fukagata, Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data, *Physics of Fluids* 32 (9) (2020) 095110. arXiv:https://pubs.aip.org/aip/pof/article-pdf/doi/10.1063/5.0020721/14710996/095110_1_online.pdf, doi:10.1063/5.0020721. URL <https://doi.org/10.1063/5.0020721>
- ⁷⁰P. Rozwood, E. Mehrez, L. Paehler, W. Sun, S. L. Brunton, Koopman-assisted reinforcement learning (2024). arXiv:2403.02290. URL <https://arxiv.org/abs/2403.02290>
- ⁷¹M. J. Carrier, W. A. Farmer, B. Srinivasan, Deep koopman neural network for analyzing high-energy-density simulations of electrical wire explosions, *IEEE Transactions on Plasma Science* 52 (10) (2024) 4916–4932. doi:10.1109/TPS.2024.3440255.
- ⁷²L. Shi, M. Haseli, G. Mamakoukas, D. Bruder, I. Abraham, T. Murphey, J. Cortes, K. Karydis, Koopman operators in robot learning (2025). arXiv:2408.04200. URL <https://arxiv.org/abs/2408.04200>
- ⁷³A. Rauh, J. Minisini, H. Aschemann, Carleman linearization for control and for state and disturbance estimation of nonlinear dynamical processes, *IFAC Proceedings Volumes* 42 (13) (2009) 455–460, 14th IFAC Conference on Methods and Models in Automation and Robotics. doi:<https://doi.org/10.3182/20090819-3-PL-3002.00079>. URL <https://www.sciencedirect.com/science/article/pii/S1474667015304821>
- ⁷⁴I. Abraham, G. de la Torre, T. Murphey, Model-based control using koopman operators, in: *Robotics: Science and Systems XIII, RSS2017, Robotics: Science and Systems Foundation*, 2017. doi:10.15607/rss.2017.xiii.052. URL <https://dx.doi.org/10.15607/RSS.2017.XIII.052>
- ⁷⁵S. E. Otto, C. W. Rowley, Koopman operators for estimation and control of dynamical systems, *Annual Review of Control, Robotics, and Autonomous Systems* 4 (Volume 4, 2021) (2021) 59–87. doi:<https://doi.org/10.1146/annurev-control-071020-010108>. URL <https://www.annualreviews.org/content/journals/10.1146/annurev-control-071020-010108>
- ⁷⁶P. Bevanda, S. Sosnowski, S. Hirche, Koopman operator dynamical models: Learning, analysis and control, *Annual Reviews in Control* 52 (2021) 197–212. doi:10.1016/j.arcontrol.2021.09.002. URL <https://dx.doi.org/10.1016/j.arcontrol.2021.09.002>
- ⁷⁷Y. Han, W. Hao, U. Vaidya, Deep learning of Koopman representation for control, in: *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 1890–1895. doi:10.1109/CDC42340.2020.9304238.
- ⁷⁸M. D. Kvalheim, S. Revzen, Existence and uniqueness of global koopman eigenfunctions for stable fixed points and periodic orbits, *Physica D: Nonlinear Phenomena* 425 (2021) 132959. doi:<https://doi.org/10.1016/j.physd.2021.132959>. URL <https://www.sciencedirect.com/science/article/pii/S0167278921001160>
- ⁷⁹M. D. Kvalheim, E. D. Sontag, Global linearization of asymptotically stable systems without hyperbolicity, *Systems & Control Letters* 203 (2025) 106163. doi:<https://doi.org/10.1016/j.sysconle.2025.106163>. URL <https://www.sciencedirect.com/science/article/pii/S0167691125001458>
- ⁸⁰O. Cornea, G. Lupton, J. Oprea, D. Tanré, Lusternik-Schnirelmann category, no. 103 in *Mathematical Surveys and Monographs*, American Mathematical Soc., 2003.
- ⁸¹P. A. Ostrand, Covering dimension in general spaces, *General Topology and its applications* 1 (3) (1971) 209–221. doi:10.1016/0016-660X(71)90093-6. URL [https://doi.org/10.1016/0016-660X\(71\)90093-6](https://doi.org/10.1016/0016-660X(71)90093-6)
- ⁸²Piecewise Carleman Embedding, <https://github.com/QuCF/QuCF/wiki/Piecewise-Carleman-embedding> (2025).