Applied Mathematics and Computation 467 (2024) 128489

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Fisher information and shape-morphing modes for solving the Fokker–Planck equation in higher dimensions

William Anderson, Mohammad Farazmand *

Department of Mathematics, North Carolina State University, 2311 Stinson Drive, Raleigh, NC 27695-8205, USA

ARTICLE INFO

Keywords:
Partial differential equations
Stochastic differential equations
Fokker–Planck equation
Neural networks
Information theory

ABSTRACT

The Fokker–Planck equation describes the evolution of the probability density associated with a stochastic differential equation. As the dimension of the system grows, solving this partial differential equation (PDE) using conventional numerical methods becomes computationally prohibitive. Here, we introduce a fast, scalable, and interpretable method for solving the Fokker–Planck equation which is applicable in higher dimensions. This method approximates the solution as a linear combination of shape-morphing Gaussians with time-dependent means and covariances. These parameters evolve according to the method of reduced-order nonlinear solutions (RONS) which ensures that the approximate solution stays close to the true solution of the PDE for all times. As such, the proposed method approximates the transient dynamics as well as the equilibrium density, when the latter exists. Our approximate solutions can be viewed as an evolution on a finite-dimensional statistical manifold embedded in the space of probability densities. We show that the metric tensor in RONS coincides with the Fisher information matrix on this manifold. We also discuss the interpretation of our method as a shallow neural network with Gaussian activation functions and time-varying parameters. In contrast to existing deep learning methods, our method is interpretable, requires no training, and automatically ensures that the approximate solution satisfies all properties of a probability density.

1. Introduction

Unlike deterministic dynamical systems, the evolution of stochastic systems cannot be unambiguously described based solely on the initial condition. Instead one studies the probability distribution of the state over time [46]. In most problems of practical interest, the evolution of the probability distribution cannot be determined analytically and therefore it needs to be approximated numerically.

For stochastic differential equations (SDEs), one can run a large ensemble of numerical simulations with different initial conditions and different realizations of the noise. Subsequently, the probability distribution of the system state can be estimated using these large-scale Monte Carlo simulations [44]. Alternatively, one can numerically solve the Fokker–Planck equation which is a partial differential equation (PDE) describing the evolution of the probability density associated with the state of the system [38].

The computational cost associated with both these approaches becomes quickly prohibitive as the dimension d of the system grows [13,33]. For instance, direct Monte Carlo methods need $\mathcal{O}(\epsilon^{-d})$ samples to reach an error tolerance $0 < \epsilon \ll 1$ [35,36,49]. On

* Corresponding author.
E-mail addresses: wmander3@ncsu.edu (W. Anderson), farazmand@ncsu.edu (M. Farazmand).

https://doi.org/10.1016/j.amc.2023.128489
Received 7 June 2023; Received in revised form 9 November 2023; Accepted 28 November 2023

the other hand, discretizing the Fokker–Planck equation requires $\mathcal{O}(N^d)$ collocation points, where N is the number of points in each direction. In either case, the computational cost grows exponentially with the dimension of the system.

We note that, for SDEs with special properties, there exist tailored methods with manageable computational cost in higher dimensions. These include SDEs with slow-fast dynamics [31] and Hamiltonian SDEs in equilibrium [47]. We refer to [13] for a review of these special cases. Nonetheless, these methods are not applicable to general SDEs and often only approximate the equilibrium density, not the transient dynamics.

The excessive computational cost of solving PDEs in higher dimensions is not specific to the Fokker–Planck equation; discretizing any PDE in higher dimensions is computationally prohibitive. Only recently, deep learning methods have been able to overcome this curse of dimensionality [22,45]. In this approach, the solution of the PDE is approximated by a deep neural network. The parameters of the network are trained so that its output solves the PDE approximately. Being mesh-free, deep learning methods are better suited for solving PDEs in higher dimensions.

There is a rapidly growing list of such deep learning methods. For instance, physics informed neural networks (PINNs) train a deep neural network by minimizing the residual of the error at prescribed collocation points [40]. Deep Galerkin method (DGM) takes a similar approach but instead of using collocation points, it minimizes the functional norm of the error [45]. Consequently, since DGM does not use collocation points, it is specially suitable for solving PDEs in higher dimensions. Another notable example is neural operators [27] which learn maps between infinite-dimensional Banach spaces and can be used to solve PDEs [30]. We refer to Beck et al. [7] for a recent review of deep learning methods for solving PDEs. Deep neural networks have already been used to solve the Fokker–Planck equation [6,14,50,54]. In spite of their impressive capabilities, these deep learning methods suffer from limited interpretability; the neural network is a black box, mapping initial-boundary conditions to the PDE's solution.

Here, we introduce an alternative method based on reduced-order nonlinear solutions (RONS). RONS approximates the solution of a PDE as a linear combination of shape-morphing modes [1–3]. In contrast to existing spectral methods, where the modes are static in time, RONS allows the modes to change shape and hence adapt to the solution of the PDE. This is achieved by allowing the modes to depend nonlinearly on a set of time-dependent shape parameters. The optimal evolution of the parameters are determined by solving a system of ordinary differential equations (ODEs), known as the RONS equation. In the case of the Fokker–Planck equation, we choose Gaussians as our shape-morphing modes. The corresponding shape parameters are the mean and covariance of each mode which are allowed to change over time in order to better approximate the solution of the PDE.

As we discuss in Section 2.1 below, our method can be interpreted as a shallow neural network. However, it has several advantages compared to existing deep learning methods:

1. Interpretability: Our method uses a linear combination of Gaussians with time-varying means and covariances. As such, the approximate solution can be easily interpreted in terms of the probability distribution of the system state. Further, a posteriori analysis of the solution is straightforward.
2. No training required: The parameters of our solution evolve according to known and computable ODEs. As a result, no training (i.e., numerical optimization) and no data are needed for determining the parameters of the network.
3. Conservation of probability: In RONS, it is straightforward to ensure that the approximate solution respects the conserved quantities of the PDE. In the case of the Fokker–Planck equation, this conserved quantity is the total probability, i.e., the integral of the probability density over the entire state space. As a result, our solutions are guaranteed to satisfy the properties of a probability density function.

Our method can easily be extended to be used with deep neural networks [17]. However, we intentionally use its shallow version to maintain its interpretability and low computational cost. The universal approximation theorem of Park and Sandberg [37] guarantees that probability densities can be approximated with such shallow neural networks to any desirable accuracy.

1.1. Original contributions

We briefly summarize our original theoretical contributions:

1. We develop the theory of shape-morphing modes for application to Fokker–Planck equations. We show that our shape-morphing approximate solutions can be interpreted as a shallow neural network with time-varying weights and biases.
2. Connection to Fisher information: We show that the metric tensor in RONS is identical to the Fisher information metric if a weighted L^2 metric is used.
3. Computational complexity: We show that symbolic RONS only requires 9 symbolic computations regardless of the dimension of the system or the number of modes used in the approximate solution.
4. Collocation RONS: When symbolic computing is not feasible, we develop a collocation point method to accurately evolve the shape-morphing solution.

In addition to the above theoretical contributions, we present four numerical examples demonstrating the feasibility, accuracy, and computational efficiency of our method.

1.2. Outline

The remainder of this paper is organized as follows. In Section 2, we review the necessary mathematical preliminaries, describe the shape-morphing solutions which approximate the Fokker–Planck equation, and discuss their interpretation as a shallow neural network. Section 3 reviews RONS for the optimal evolution of the shape parameters. In Section 4, we discuss the relationship between our method and the Fisher information metric. The performance of our method is demonstrated on several numerical examples in Section 5. Finally, we present our concluding remarks in Section 6.

2. Mathematical preliminaries

Consider the Itô stochastic differential equation,

$$\mathrm{d}\mathbf{X} = \mathbf{F}(\mathbf{X}, t)\mathrm{d}t + \sigma \mathrm{d}\mathbf{W}, \quad \mathbf{X}(0) = \mathbf{X}_0 \tag{1}$$

where $\mathbf{X}(t) \in \mathbb{R}^d$ denotes the state vector at time $t \geq 0$, $\mathbf{F} : \mathbb{R}^d \to \mathbb{R}^d$ is a sufficiently smooth vector field, and $\mathbf{W}(t)$ is the standard Wiener process in $\mathbb{R}^d$ with intensity $\sigma > 0$. The initial condition $\mathbf{X}_0$ can itself be uncertain and drawn randomly from a probability density $p_0(\mathbf{x})$. Although here we only consider the homogeneous additive noise, the following framework can easily be extended to the case of inhomogeneous multiplicative noise where the noise intensity matrix $\sigma(\mathbf{X}, t) \in \mathbb{R}^{d\times d}$ depends on the state.

The probability density $p(\mathbf{x}, t)$ corresponding to the SDE (1) satisfies the Fokker–Planck equation,

$$\frac{\partial p}{\partial t} = \mathcal{L}p := -\nabla \cdot [\mathbf{F}p] + \nu \Delta p, \quad p(\mathbf{x}, 0) = p_0(\mathbf{x}), \tag{2}$$

where the diffusion coefficient is given by $\nu = \sigma^2/2$ and $\mathcal{L}$ is a linear differential operator which depends on the vector field $\mathbf{F}(\mathbf{x}, t)$. Being a probability density, the solution p is non-negative and belongs to the Lebesgue space $L^1(\mathbb{R}^d)$ [26]. Furthermore, the norm of the solution in this space is conserved and equal to unity, i.e.,

$$\|p(\cdot, t)\|_{L^1} = 1, \tag{3}$$

for all time $t \geq 0$.

As discussed in the Introduction, when the dimension d is large, discretizing the Fokker–Planck equation becomes computationally prohibitive. To overcome this curse of dimensionality, we use a mesh-free method by considering an approximate solution of the form,

$$\hat{p}(\mathbf{x}, \boldsymbol{\theta}(t)) = \sum_{i=1}^{r} A_i^2(t) \exp\left[-\frac{|\mathbf{x} - \mathbf{c}_i(t)|^2}{L_i^2(t)}\right], \tag{4}$$

which is a sum of Gaussians. Here $A_i^2(t) \in \mathbb{R}^+$ is the amplitude of the i-th mode, $L_i^2(t) \in \mathbb{R}^+$ is its variance, and $\mathbf{c}_i(t) \in \mathbb{R}^d$ is its mean. The collection of amplitudes, variances and means constitutes the time-dependent shape parameters $\boldsymbol{\theta} = \left\{A_i, L_i, \mathbf{c}_i\right\}_{i=1}^{r}$. Therefore, the approximate solution contains a total of $n = r(d + 2)$ parameters.

The key to the success of our method is the fact that the parameters $\boldsymbol{\theta}(t)$ are allowed to change over time. This enables the modes in the approximate solution (4) to change their shape and location, hence adapting to the solution of the PDE. Of course, the immediate question is how to evolve the parameters $\boldsymbol{\theta}(t)$. As we review in Section 3, RONS evolves the parameters according to a set of ordinary differential equations. These ODEs are designed such that the solution $\hat{p}(\mathbf{x}, \boldsymbol{\theta}(t))$ approximates the dynamics of the Fokker–Planck equation. But before describing RONS, we first provide the justification for using Gaussian modes in the approximate solution (4).

2.1. Choice of the modes

The approximate solution (4) consists of a sum of Gaussians. In general, other nonlinear functions can be used as modes [1]. However, the Gaussian seems appropriate for the Fokker–Planck equation. First, note that if the vector field $\mathbf{F}(\mathbf{x}, t)$ is linear in $\mathbf{x}$, the stationary solution to the corresponding Fokker–Planck equation will be a Gaussian [46]. More importantly, the following universal approximation theorem guarantees that any function in $L^1(\mathbb{R}^d)$ can be approximated, to any desirable accuracy, with a function of the form (4).

Theorem 2.1 *(Park and Sandberg [37]). Let $K : \mathbb{R}^d \to \mathbb{R}$ be integrable, bounded and continuous almost everywhere. If $\int_{\mathbb{R}^d} K(\mathbf{x})\mathrm{d}\mathbf{x} \neq 0$, then the set*

$$\left\{ \sum_{i=1}^{r} A_i K\left(-\frac{|\mathbf{x} - \mathbf{c}_i|^2}{L_i^2}\right) : A_i \in \mathbb{R}, L_i \neq 0, \mathbf{c}_i \in \mathbb{R}^d, r \in \mathbb{N} \right\} \tag{5}$$

is dense in $L^q(\mathbb{R}^d)$ for all $1 \leq q < \infty$.

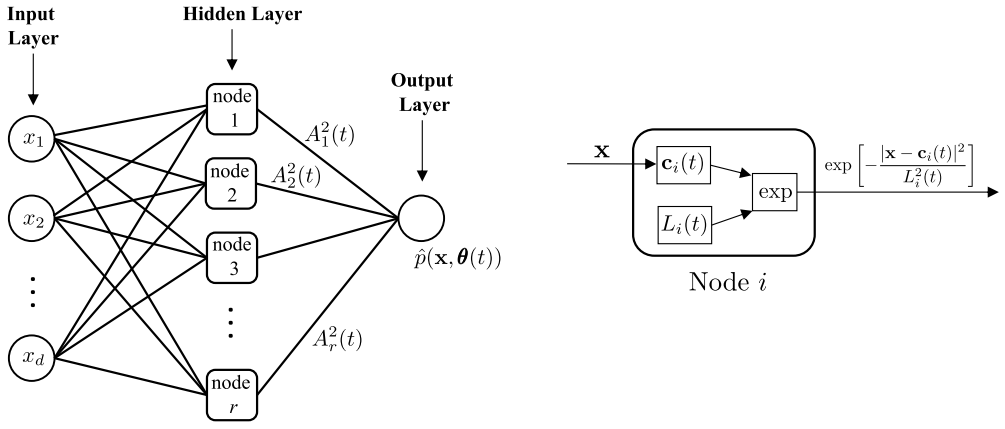

Fig. 1. Interpretation of the approximate solution (4) as a shallow neural network. Left: Network architecture. Right: Internal structure of each node.

The Gaussian function clearly satisfies all the conditions in Theorem 2.1. Therefore, any probability density function p in $L^1(\mathbb{R}^d)$ can be approximated, to arbitrary accuracy, with a function $\hat{p}$ of the form (4). More precisely, given $p \in L^1(\mathbb{R}^d)$, for any $\epsilon > 0$, there exist shape parameters $\boldsymbol{\theta}$ and $r \in \mathbb{N}$ such that $\|p - \hat{p}\|_{L^1} < \epsilon$. We point out that the amplitudes A_i in the approximate solution (4) are squared to ensure that $\hat{p}$ is non-negative as is required for a probability density.

Note that the covariance matrix of each Gaussian in (4) is diagonal. One could alternatively choose a non-diagonal covariance matrix $\Sigma_i(t) \in \mathbb{R}^{d\times d}$ and use the modes,

$$\exp\left[-\frac{1}{2}(\mathbf{x}-\mathbf{c}_i)^\top \Sigma_i^{-1}(t)(\mathbf{x}-\mathbf{c}_i)\right]. \tag{6}$$

In higher dimensions, this would significantly increase the number of shape parameters per mode since the covariance matrix needs to be solved for simultaneously. Fortunately, Theorem 2.1 implies that diagonal covariance matrices are sufficient for approximating any probability density as long as the number of modes r is large enough.

We point out that the approximate solution (4) can be viewed as a shallow artificial neural network with one hidden layer. As depicted in Fig. 1, this network takes the coordinates $\mathbf{x} = (x_1, x_2, \cdots, x_d)$ as inputs and returns $\hat{p}(\mathbf{x}, \boldsymbol{\theta}(t))$ as output. The activation function of each node is a Gaussian with parameters $L_i(t)$ and $\mathbf{c}_i(t)$. The amplitudes $A_i^2(t)$ are the output weights of the neural network. This network differs from conventional neural networks in that its parameters are time-dependent. As such, it is a special type of an evolutional neural network first introduced in [17].

Furthermore, in conventional neural networks, the parameters are determined through the so-called training process where the parameters are iteratively tuned to match the training data. In contrast, here we use RONS to evolve the parameters so that the approximate solution $\hat{p}(\mathbf{x}, \boldsymbol{\theta}(t))$ matches the dynamics of Fokker–Planck as closely as possible. This method, which requires no training and no data, is described in Section 3 below.

3. Evolution of parameters

RONS evolves the shape parameters $\boldsymbol{\theta}(t)$ such that the discrepancy between the approximate solution (4) and the dynamics of the PDE (2) is instantaneously minimized [1]. Here, we briefly review this method in the context of Fokker–Planck equation and refer to Anderson and Farazmand [1,3] for further detail.

Note that since $\hat{p}$ is an approximate solution $\partial_t \hat{p}$ does not necessarily coincide with the right-hand side $\mathcal{L}\hat{p}$ of the Fokker–Planck equation. We define the residual

$$R(\mathbf{x}; \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \sum_{j=1}^{n} \frac{\partial \hat{p}}{\partial \theta_j}(\mathbf{x}, \boldsymbol{\theta})\dot{\theta}_j - \mathcal{L}\hat{p}(\mathbf{x}, \boldsymbol{\theta}), \tag{7}$$

where we used the fact that

$$\frac{\partial}{\partial t}\hat{p}(\mathbf{x}, \boldsymbol{\theta}(t)) = \sum_{j=1}^{n} \frac{\partial \hat{p}}{\partial \theta_j}(\mathbf{x}, \boldsymbol{\theta})\dot{\theta}_j. \tag{8}$$

RONS determines the evolution of the parameters $\boldsymbol{\theta}(t)$ by minimizing the residual R in an appropriate sense. We consider two options: 1) Symbolic RONS which minimizes a functional norm of the residual, and 2) Collocation RONS which minimizes the residual at prescribed collocation points. For completeness, we review each approach in Sections 3.1 and 3.2 below.

3.1. Symbolic RONS

For a fixed $\boldsymbol{\theta}$ and $\dot{\boldsymbol{\theta}}$, consider $R(\mathbf{x};\boldsymbol{\theta},\dot{\boldsymbol{\theta}})$ as a map from $\mathbf{x}$ to the real line, i.e., $R(\cdot;\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) : \mathbb{R}^d \to \mathbb{R}$. We assume that this map belongs to a Hilbert space H with the inner product $\langle\cdot,\cdot\rangle_H$ and the induced norm $\|\cdot\|_H$. Here, we describe RONS for a general Hilbert space and in Section 4 we identify specific choices of the Hilbert space suitable for the Fokker–Planck equation.

In symbolic RONS, we minimize the residual norm $\|R(\cdot;\boldsymbol{\theta},\dot{\boldsymbol{\theta}})\|_H$ with the constraint that

$$I(\boldsymbol{\theta}) := \|\hat{p}(\cdot,\boldsymbol{\theta})\|_{L^1} = 1. \tag{9}$$

For the approximate solution (4), we have $I(\boldsymbol{\theta}) = \sum_{i=1}^r A_i^2(\pi L_i^2)^{d/2}$. This constraint is required to ensure that the approximate solution $\hat{p}$ is a probability density function. The resulting constrained optimization problem reads,

$$\min_{\dot{\boldsymbol{\theta}}} \|R(\cdot;\boldsymbol{\theta},\dot{\boldsymbol{\theta}})\|_H^2 + \alpha|\dot{\boldsymbol{\theta}}|^2, \tag{10a}$$

$$\text{such that} \quad I(\boldsymbol{\theta}) = 1, \tag{10b}$$

where $\alpha \geq 0$ is a Tikhonov regularization parameter and $|\cdot|$ denotes the usual Euclidean norm. In the absence of regularization ($\alpha = 0$), equation (10a) minimizes the instantaneous discrepancy between the rate of change of the approximate solution $\partial_t\hat{p}$ and the rate of change $\mathcal{L}\hat{p}$ dictated by the PDE. The motivation for the Tikhonov regularization will become clear in Section 3.1.1. One may also formulate a finite-time version of the constrained optimization problem (10). However, this finite-time formulation tends to return unstable equations for the evolution of parameters $\boldsymbol{\theta}(t)$ (cf. Appendix A of [1]).

As shown in [1,3], the solution to the constrained optimization problem (10) satisfies the system of ordinary differential equations,

$$\left[M(\boldsymbol{\theta}) + \alpha\mathbb{I}_n\right]\dot{\boldsymbol{\theta}} = \mathbf{f}(\boldsymbol{\theta}) - \lambda\nabla I(\boldsymbol{\theta}), \tag{11}$$

where the gradient is taken with respect to the parameters $\boldsymbol{\theta}$, and $\mathbb{I}_n$ denotes the $n\times n$ identity matrix with $n = r(d+2)$ being the total number of parameters. We refer to equation (11) as symbolic RONS equation, or S-RONS for short. The motivation for the term *symbolic* will become clear at the end of this section.

The metric tensor $M(\boldsymbol{\theta})$ is a symmetric positive semi-definite matrix whose entries are given by

$$M_{ij} = \left\langle \frac{\partial\hat{p}}{\partial\theta_i}, \frac{\partial\hat{p}}{\partial\theta_j} \right\rangle_H, \quad i,j \in \{1,2,\cdots,n\}. \tag{12}$$

Note that, since the metric tensor is symmetric positive semi-definite, the matrix $M(\boldsymbol{\theta}) + \alpha\mathbb{I}_n$ is invertible for all $\alpha > 0$. The vector field $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ is defined by

$$f_i = \left\langle \frac{\partial\hat{p}}{\partial\theta_i}, \mathcal{L}\hat{p} \right\rangle_H, \quad i = 1,2,\cdots,n. \tag{13}$$

Finally, $\lambda \in \mathbb{R}$ is a Lagrange multiplier defined by

$$\lambda = \frac{\langle\nabla I, (M+\alpha\mathbb{I}_n)^{-1}\mathbf{f}\rangle}{\langle\nabla I, (M+\alpha\mathbb{I}_n)^{-1}\nabla I\rangle}, \tag{14}$$

where $\langle\cdot,\cdot\rangle$ denotes the usual Euclidean inner product.

As shown in [3], in the absence of regularization ($\alpha = 0$), the ODEs (11) become stiff as the number of parameters n grows. As a result, solving the ODEs using explicit integration schemes requires exceedingly small time steps. However, as we show in Section 5 below, even small values of the regularization parameter $\alpha > 0$ alleviate this stiffness issue.

3.1.1. Symbolic computing for S-RONS

Finally, we comment on the computational cost of the symbolic RONS equation (11). This equation requires computing functional inner products $\langle\cdot,\cdot\rangle_H$ over the Hilbert space H. When feasible, we use symbolic computing to obtain closed-form symbolic expressions for the metric tensor M_{ij} and the right-hand side vector field f_i; hence the name symbolic RONS. This is desirable since symbolic expressions evade quadrature errors. Furthermore, the inner products do not need to be recomputed during time integration; the existing symbolic expressions are evaluated by substituting the updated values of the parameters $\boldsymbol{\theta}(t)$.

However, as the number of parameters $n = r(d+2)$ increases, brute-force computation of the inner products becomes expensive. Computing the metric tensor, for instance, requires $\mathcal{O}(n^2)$ symbolic computations. Fortunately, as discussed in [3], the special structure of the metric tensor implies that a far smaller number of symbolic computations are required. More specifically, the entire metric tensor $M \in \mathbb{R}^{n\times n}$ can be evaluated by computing only 6 symbolic expressions. Similarly, the right-hand side vector field $\mathbf{f} \in \mathbb{R}^n$ can be evaluated by computing only 3 symbolic expressions. Therefore, to form the entire S-RONS equation (11), only 9 symbolic computation of inner products is needed. Note that this number is independent of the number of terms r in the approximate solution (4) and the dimension d of the SDE. As such, S-RONS is scalable to higher dimensions and the number of terms can be increased to achieved a desired accuracy at no significant additional computational cost.

To better understand this low computational cost, note that the inner products in the metric tensor (12) involve the terms,

$$\frac{\partial \hat{p}}{\partial A_k} = 2A_k \exp\left[-\frac{|\mathbf{x}-\mathbf{c}_k|^2}{L_k^2}\right], \tag{15a}$$

$$\frac{\partial \hat{p}}{\partial L_k} = \frac{2A_k^2}{L_k^3}|\mathbf{x}-\mathbf{c}_k|^2 \exp\left[-\frac{|\mathbf{x}-\mathbf{c}_k|^2}{L_k^2}\right], \tag{15b}$$

$$\frac{\partial \hat{p}}{\partial c_{k,\ell}} = \frac{2A_k^2}{L_k^2}\left(x_\ell - c_{k,\ell}\right) \exp\left[-\frac{|\mathbf{x}-\mathbf{c}_k|^2}{L_k^2}\right], \tag{15c}$$

where $1 \le k \le r$, $1 \le \ell \le d$, and $c_{k,\ell}$ denotes the ℓ-th component of the vector $\mathbf{c}_k$. Assume that we have computed symbolic expressions for the inner products $\langle \cdot, \cdot \rangle_H$ between the terms in equation (15) with general indices (k, ℓ). Then the entire metric tensor M can be evaluated by substituting the numerical values of A_k, L_k, and $c_{k,\ell}$ in the computed symbolic expressions. Therefore, only 6 symbolic expressions need to be computed to populate the entire matrix M.

Similarly, to compute the right-hand side vector (13), we only need to compute symbolic expressions for the inner product of $\mathcal{L}\hat{p}$ with the three terms in equation (15). Therefore, the entire vector $\mathbf{f}(\theta)$ can be evaluated using 3 symbolic expressions.

This observation leads to enormous computational savings as the number of dimensions and/or the number of terms in the approximate solution increase. For instance, in $d = 5$ dimensions and with $r = 10$ terms in the approximate solution, brute-force computation of the S-RONS equation would require symbolic computation of $n(n+3)/2 = 2,555$ expressions, taking into account that the metric tensor M is symmetric. However, as discussed above, in reality only 9 symbolic expressions need to be computed in symbolic RONS.

We emphasize that symbolic computing reduces the computational cost of the time integration as well. Since all terms are computed symbolically, as time integration progresses, these terms do not need to be recomputed; instead, they will be evaluated by substituting the new values of $\boldsymbol{\theta}(t)$ into existing symbolic expressions.

For certain choices of the Hilbert space H, existing symbolic computing software are unable to return a closed-form expression for the metric tensor or the right-hand side vector field. Collocation RONS addresses this issue.

3.2. Collocation RONS

Symbolic computation of the RONS terms in (11) may not be straightforward for certain Hilbert spaces H and certain vector fields $\mathbf{F}(\mathbf{x}, t)$. Collocation RONS, or C-RONS for short, was developed in [3] to address this issue. In this approach, no functional inner products need to be computed and therefore the computational cost of forming the RONS equations reduces drastically.

In C-RONS, instead of minimizing the residual R over the entire state space $\mathbb{R}^d$, we minimize it over a set of prescribed collocation points $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$. More precisely, we solve the constrained optimization problem,

$$\min_{\dot{\boldsymbol{\theta}}} \sum_{i=1}^{N} |R(\mathbf{x}_i; \boldsymbol{\theta}, \dot{\boldsymbol{\theta}})|^2 + \alpha |\dot{\boldsymbol{\theta}}|^2, \tag{16a}$$

$$\text{such that} \quad I(\boldsymbol{\theta}) = 1. \tag{16b}$$

In the absence of regularization ($\alpha = 0$), this optimization problem minimizes the sum of squares of the residual at the collocation points given the constraint that $I(\boldsymbol{\theta}) = \|\hat{p}(\cdot, \boldsymbol{\theta})\|_{L^1} = 1$. As in S-RONS, the regularization ensures that the resulting ODEs are not stiff and therefore can be integrated in time using explicit discretization schemes.

As shown in [3], the solution to the optimization problem (16) satisfies the system of ODEs,

$$\left[\widetilde{M}(\boldsymbol{\theta}) + \alpha \mathbb{I}_n\right] \dot{\boldsymbol{\theta}} = \widetilde{\mathbf{f}}(\boldsymbol{\theta}) - \widetilde{\lambda} \nabla I(\boldsymbol{\theta}), \tag{17}$$

where the collocation metric tensor is given by

$$\widetilde{M}(\boldsymbol{\theta}) = J^\top J, \quad J_{ij}(\boldsymbol{\theta}) = \frac{\partial \hat{p}}{\partial \theta_j}(\mathbf{x}_i, \boldsymbol{\theta}), \quad i \in \{1, 2, \cdots, N\}, \quad j \in \{1, 2, \cdots, n\}. \tag{18}$$

The right-hand vector field $\widetilde{\mathbf{f}} : \mathbb{R}^n \to \mathbb{R}^n$ is given by

$$\widetilde{\mathbf{f}}(\boldsymbol{\theta}) = J^\top \mathbf{f}(\boldsymbol{\theta}), \quad f_i(\boldsymbol{\theta}) = \mathcal{L}\hat{p}\Big|_{\mathbf{x}=\mathbf{x}_i}, \quad i \in \{1, 2, \cdots, N\}, \tag{19}$$

and the Lagrange multiplier is defined by

$$\widetilde{\lambda} = \frac{\langle \nabla I, [\widetilde{M}(\boldsymbol{\theta}) + \alpha \mathbb{I}_n]^{-1} \widetilde{\mathbf{f}} \rangle}{\langle \nabla I, [\widetilde{M}(\boldsymbol{\theta}) + \alpha \mathbb{I}_n]^{-1} \nabla I \rangle}. \tag{20}$$

We refer to equation (17) as the C-RONS equation. Note that, unlike S-RONS, forming the C-RONS equation does not require computing any functional inner products; it only needs point-wise evaluation at the collocation points $\mathbf{x}_i$. Therefore, the C-RONS equation can be formed at a lower computational cost. However, this lower computational cost comes at the expense of accuracy

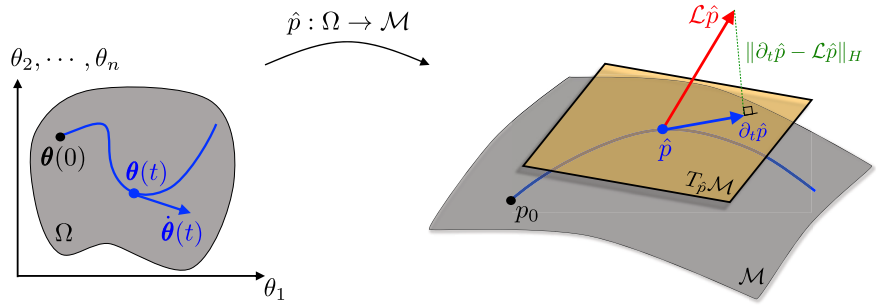

Fig. 2. Geometric illustration of the shape-morphing approximate solutions. The image of $\hat{p}$ is a statistical manifold $\mathcal{M} \subset L^1(\mathbb{R}^d)$. Any parameter value $\boldsymbol{\theta}$ defines a point $\hat{p}(\cdot, \boldsymbol{\theta})$ on this manifold, with the corresponding tangent space $T_{\hat{p}}\mathcal{M}$.

since only the residual error at the collocation points is minimized, whereas S-RONS minimizes the error over the entire state space. Furthermore, a straightforward implementation of C-RONS is not scalable to higher dimensions. For instance, if the collocation points are chosen uniformly with N points in each direction, C-RONS would require $\mathcal{O}(N^d)$ collocation points. Although this number can be reduced by choosing the collocation points adaptively [9], it cannot outperform S-RONS which requires only 9 symbolic computations regardless of the spatial dimension.

We conclude this section by commenting on the special case of S-RONS where the Hilbert space H is the space of square integrable functions $L^2(\mathbb{R}^d)$ and no regularization is used ($\alpha = 0$). Assume that we approximate the inner products (12) and (13) using Monte Carlo integration instead of symbolic computing, where the Monte Carlo samples $\mathbf{x}_i \in \mathbb{R}^d$ are drawn at random. In this case, the S-RONS equation (11) coincides with the C-RONS equation (17). We refer to [3] for the equivalence proof. Du and Zaki [17] used this Monte Carlo approximation with the collocation points distributed according to the uniform Lebesgue measure on $\mathbb{R}^d$. Bruna et al. [9] proposed an adaptive sampling method where they draw their collocation points from a distribution which evolves in time. We point out that neither [17] nor [9] use regularization or ensure preservation of conserved quantities.

4. Choice of the metric

Recall that in symbolic RONS, the residual error is minimized with respect to the norm $\|\cdot\|_H$ defined on the Hilbert space H. So far, we have stated the results for a general Hilbert space. In this section, we determine the specific choice of the Hilbert space for the Fokker–Planck equation.

To this end, we first view the approximate solution (4) as a map from the parameters $\boldsymbol{\theta} \in \mathbb{R}^n$ to the space of probability densities $L^1(\mathbb{R}^d)$,

$$\begin{aligned} \hat{p} : \Omega &\to L^1(\mathbb{R}^d) \\ \boldsymbol{\theta} &\mapsto \hat{p}(\cdot, \boldsymbol{\theta}), \end{aligned} \tag{21}$$

where $\Omega \in \mathbb{R}^n$ is the set of all admissible parameters $\boldsymbol{\theta}$. As illustrated in Fig. 2, the image of the map $\hat{p}$ forms an n-dimensional subset of the infinite-dimensional function space $L^1(\mathbb{R}^d)$. In fact, under certain assumptions, the image of $\hat{p}$ is an immersed manifold $\mathcal{M}$ [1]. Note that, although the parameter $\boldsymbol{\theta}$ is a finite-dimensional vector, the image $\hat{p}(\cdot, \boldsymbol{\theta})$ is a function of $\mathbf{x}$, belonging to the infinite-dimensional space $L^1(\mathbb{R}^d)$.

The set $\mathcal{M}$ is a statistical manifold in the sense that every point on it is a probability density function [34]. The intrinsic metric associated with a statistical manifold is the so-called Fisher information metric [20]. More specifically, the metric tensor associated with the Fisher information metric is given by

$$g_{ij}(\boldsymbol{\theta}) = \int_{\mathbb{R}^d} \frac{\partial \log \hat{p}}{\partial \theta_i} \frac{\partial \log \hat{p}}{\partial \theta_j} \hat{p} \, \mathrm{d}\mathbf{x} = \int_{\mathbb{R}^d} \frac{1}{\hat{p}(\mathbf{x}, \boldsymbol{\theta})} \frac{\partial \hat{p}}{\partial \theta_i}(\mathbf{x}, \boldsymbol{\theta}) \frac{\partial \hat{p}}{\partial \theta_j}(\mathbf{x}, \boldsymbol{\theta}) \mathrm{d}\mathbf{x}. \tag{22}$$

Measuring the distance between two probability distributions $\hat{p}(\cdot, \boldsymbol{\theta}_1)$ and $\hat{p}(\cdot, \boldsymbol{\theta}_2)$ according to the metric tensor (22) returns the Fisher–Rao distance between parameterized probability distributions [41,10].

Therefore, a suitable metric defined on the statistical manifold $\mathcal{M}$ is the Fisher information metric. Next, we return to the RONS equation (11). Recall that we stated the results for a general Hilbert space H. Now, let us consider the specific Hilbert space $H = L^2_\mu(\mathbb{R}^d)$, where $\mu = \hat{p}^{-1}\mathrm{d}\mathbf{x}$ is a weighted Lebesgue measure. In this case, the metric tensor (12) can be written explicitly as

$$M_{ij}(\boldsymbol{\theta}) = \left\langle \frac{\partial \hat{p}}{\partial \theta_i}, \frac{\partial \hat{p}}{\partial \theta_j} \right\rangle_{L^2_\mu} = \int_{\mathbb{R}^d} \frac{1}{\hat{p}} \frac{\partial \hat{p}}{\partial \theta_i} \frac{\partial \hat{p}}{\partial \theta_j} \, \mathrm{d}\mathbf{x}. \tag{23}$$

We notice that using the Hilbert space $H = L^2_\mu(\mathbb{R}^d)$, the RONS metric tensor M_{ij} coincides with Fisher information metric (22). In other words, taking $H = L^2_\mu(\mathbb{R}^d)$ induces the Fisher information matrix on the manifold $\mathcal{M}$ defined by the RONS approximate solution $\hat{p}$. Therefore, for the Fokker–Planck equation, a suitable choice of the Hilbert space is the weighted Lebesgue space $L^2_\mu(\mathbb{R}^d)$.

We point out that Bruna et al. [9] had already proposed this weighted Lebesgue space in an ad hoc manner in their adaptive Monte Carlo estimation of the integrals (12). It is interesting that this adaptive sampling can be rigorously justified in the case of Fokker–Planck equation.

In our experience, using symbolic computation to obtain a closed-form expression for (23) is not always feasible. In such cases, we use the Hilbert space $H = L^2(\mathbb{R}^d)$ to ensure symbolic computing is feasible at the cost of sacrificing the connection between RONS and the Fisher information metric. In Section 5.2, we discuss the ramification of this trade-off on a specific example.

5. Numerical results and discussion

In this section, we assess the accuracy and computational cost of our method on a number of SDEs with progressively higher level of complexity. In all cases, we use Mathematica for symbolic computing and Matlab for numerical time integration of the RONS equations.

5.1. Benchmark example: Ornstein–Uhlenbeck process

We consider a one-dimensional (1D) Ornstein–Uhlenbeck (OU) process and show that a Gaussian evolved according to RONS coincides exactly with the true solution of the Fokker–Planck equation corresponding to the OU process.

The Ornstein-Uhlenbeck process $X(t)$ satisfies the SDE,

$$\mathrm{d}X = -\gamma X \mathrm{d}t + \sigma \mathrm{d}W, \quad X(0) = 0, \tag{24}$$

where $\gamma > 0$ is the drift coefficient and $\sigma > 0$ is noise intensity. The Fokker–Planck equation associated with the OU process (24) is

$$\frac{\partial p}{\partial t} = \gamma \frac{\partial}{\partial x}\left(xp\right) + \frac{\sigma^2}{2}\frac{\partial^2 p}{\partial x^2}, \quad p\,(x,0) = \delta\,(x), \tag{25}$$

where $\delta(x)$ is the Dirac delta function centered at the origin. The Fokker–Planck equation (25) admits the exact solution [23],

$$p(x,t) = \sqrt{\frac{\gamma}{\pi\sigma^2(1-\exp[-2\gamma t])}}\exp\left[-\frac{\gamma x^2}{\sigma^2(1-\exp[-2\gamma t])}\right], \tag{26}$$

which is a Gaussian whose amplitude decays over time while its variance grows. Also note that $p(x,t)$ tends to the Dirac delta function $\delta(x)$ as time t tends to zero.

To apply RONS, we consider the Gaussian solution,

$$\hat{p}(x,\boldsymbol{\theta}(t)) = A(t)\exp\left[-\frac{(x-c(t))^2}{L^2(t)}\right], \tag{27}$$

with the time-dependent parameters $\boldsymbol{\theta}(t) = (A(t), L(t), c(t))^\top$. Note that this is the Gaussian approximate solution (4) with only one mode ($r = 1$). Here, we do not square the amplitude to simplify the following analysis.

As mentioned in Section 2, the approximate solution $\hat{p}$ is a probability density function (PDF), and so we must ensure that total probability of our approximate solution is always equal to one. This is a conserved quantity of the Fokker–Planck equation which we enforce when applying RONS by ensuring

$$I(\boldsymbol{\theta}(t)) = \int_{\mathbb{R}} \hat{p}(x,\boldsymbol{\theta}(t))\,\mathrm{d}x = \sqrt{\pi}A(t)L(t) = 1, \quad \forall t \geq 0. \tag{28}$$

Applying RONS to the Fokker–Planck equation associated with the OU process, using the Gaussian approximate solution (27) and the Hilbert space $H = L^2(\mathbb{R})$, the corresponding S-RONS equation (11) reads

$$\dot{A} = A\left(\gamma - \frac{\sigma^2}{L^2}\right), \quad \dot{L} = \frac{\sigma^2}{L} - \gamma L, \quad \dot{c} = -\gamma c. \tag{29}$$

To solve these ODEs, we need to specify the appropriate initial conditions. For the Gaussian (27) to coincide with the initial condition of the Fokker–Planck equation (25) at time $t = 0$, we choose the initial parameter values,

$$A(t_0) = \sqrt{\frac{\gamma}{\pi\sigma^2(1-\exp[-2\gamma t_0])}}, \quad L(t_0) = \frac{1}{\sqrt{\pi}A(t_0)}, \quad c(t_0) = 0. \tag{30}$$

Note that as $t_0 \to 0$, the initial condition $\hat{p}(\mathbf{x},\boldsymbol{\theta}(t_0))$ approaches the Dirac delta function $\delta(x)$ as required. The exact solution to the S-RONS equation (29), with the initial condition (30), is given by

$$A(t) = \sqrt{\frac{\gamma}{\pi\sigma^2(1-\exp[-2\gamma t])}}, \quad L(t) = \left(\frac{\gamma}{\sigma^2(1-\exp[-2\gamma t])}\right)^{-1/2}, \quad c(t) = 0. \tag{31}$$

Substituting this solution into $\hat{p}(\mathbf{x},\boldsymbol{\theta}(t))$, we recover the exact solution (26) to the Fokker–Planck equation. This benchmark example shows that the RONS solution coincides with the exact solution of the Fokker–Planck equation for the OU process.

5.2. One-dimensional bistable potential

In this section, we consider an SDE where the dynamics are driven by a potential function. Our main focus here is to highlight the impact of the choice of the Hilbert space H and the regularization parameter α. We consider the SDE

$$\mathrm{d}X(t) = -V'(x)\,\mathrm{d}t + \sigma \mathrm{d}W, \tag{32}$$

where $V : \mathbb{R} \to \mathbb{R}$ is the potential. The Fokker–Planck equation corresponding to (32) reads

$$\frac{\partial p}{\partial t} = \frac{\partial}{\partial x}\big(V'(x)p\big) + \nu \frac{\partial^2 p}{\partial x^2}. \tag{33}$$

In general, the analytical solution for this Fokker-Plank equation for all times is not known. However, the asymptotically stable steady state solution is given by

$$p_{\mathrm{eq}}(x) = C \exp\left[-\frac{V(x)}{\nu}\right], \tag{34}$$

where C is a normalizing constant.

Here, we consider the potential

$$V(x) = \frac{x^4}{4} - \frac{x^2}{2}. \tag{35}$$

This potential is symmetric with two minima at $x = \pm 1$, and so the equilibrium solution (34) is bimodal with peaks at $x = \pm 1$. Similar bistable potentials have been studied by several other authors [8,19,32,53].

To apply RONS we once again consider the Gaussian ansatz (4), where we will now use sums of Gaussians rather than a single Gaussian in our approximation. As before, we also enforce that total probability of the approximate solution is always equal to 1 to ensure that $\hat{p}$ is in fact a probability density function.

In addition to exploring the effects from changing the number of modes used in our approximate solution, we also study the choice of Hilbert space H for our inner products. In particular, we compare the results for the Hilbert space $H = L^2(\mathbb{R})$ and the weighted Hilbert space $H = L^2_\mu(\mathbb{R})$. As discussed in Section 4, when using the weighted inner product, the metric tensor M coincides with the Fisher information matrix.

Using the Hilbert space $H = L^2(\mathbb{R})$, we are able to symbolically calculate the inner products of the RONS equations and apply S-RONS. This approach is scalable and allows for rapid time integration of the ODEs. In contrast, when using the Hilbert space $H = L^2_\mu(\mathbb{R})$, obtaining closed-form symbolic expressions for the inner products was not possible. Consequently, we resort to using C-RONS for the weighted L^2 inner product space. Note that C-RONS requires sampling which can be expensive in higher dimensions, but it is not an obstacle in this 1D problem.

For this section, we integrate the RONS equation using Matlab's `ode15s` solver [43]. In our numerical experiments, `ode15s` takes large time steps once the approximate solution $\hat{p}$ is near its equilibrium solution $\hat{p}_{\mathrm{eq}}$. This allows for rapid simulations over long time scales, which helps us obtain the equilibrium solution predicted by RONS.

We first apply RONS to the Fokker–Planck equation (33) with the bistable potential, using $r = 2$ modes for the Gaussian approximate solution (4). We use a noise intensity of $\sigma = 0.5$ for all simulations in this section. Fig. 3 compares the evolution of the approximate solution $\hat{p}$ predicted by S-RONS using the L^2 inner product and C-RONS using the weighted L^2_μ inner product. When applying C-RONS we take 100 equidistant collocation points on the interval $x = [-4, 4]$.

First we observe that using the Hilbert space $H = L^2_\mu(\mathbb{R})$ produces an approximate solution which is a reasonable approximation of the equilibrium density by two Gaussians. However, taking $H = L^2(\mathbb{R})$ fails in this case as both Gaussians converge to the same peak at $x = -1$. We have observed the same behavior for a wide range of initial conditions for the two Gaussians. When using $H = L^2_\mu(\mathbb{R})$, corresponding to the Fisher information metric, the approximate solution always converges to the true equilibrium density, whereas using $H = L^2(\mathbb{R})$ with two Gaussians leads to the incorrect equilibrium. The only exception to this is when we start from initial conditions which are symmetric, with the Gaussians initially placed on the opposite sides of the origin. In this particular case, the approximate solution converges to a reasonable approximation of the true equilibrium even when we use $H = L^2(\mathbb{R})$ (not shown here).

The fact that using the weighted inner product space $H = L^2_\mu(\mathbb{R})$ leads to better results is not surprising. As discussed in Section 4, this is equivalent to using the Fisher information metric on the manifold of the approximate solution which is the natural metric for a statistical manifold. Nonetheless, using the unweighted Hilbert space $L^2(\mathbb{R})$ is still desirable since it allows us to use S-RONS which requires no sampling and incurs no numerical error in approximating the inner products. So, can we somehow fix the issue of converging to the wrong equilibrium solution using $H = L^2(\mathbb{R})$? The answer is yes as long as the number of terms r in the approximate solution (4) is large enough.

For instance, let us consider the same initial condition used to produce Fig. 3, but we now use $r = 10$ Gaussians in our approximate solution. As discussed in Section 3, when using a large number of parameters in the approximate solution, we must apply Tikhonov regularization to alleviate the stiffness of the RONS ODEs. Fig. 4(a) shows the evolution of the approximate solution $\hat{p}$ with $H = L^2(\mathbb{R})$ and the regularization parameter $\alpha = 10^{-4}$. Unlike the previous case where only two Gaussians were used, the approximate solution converges to the correct equilibrium density. Of course, one should be cautious not to over-regularize the problem. As shown in

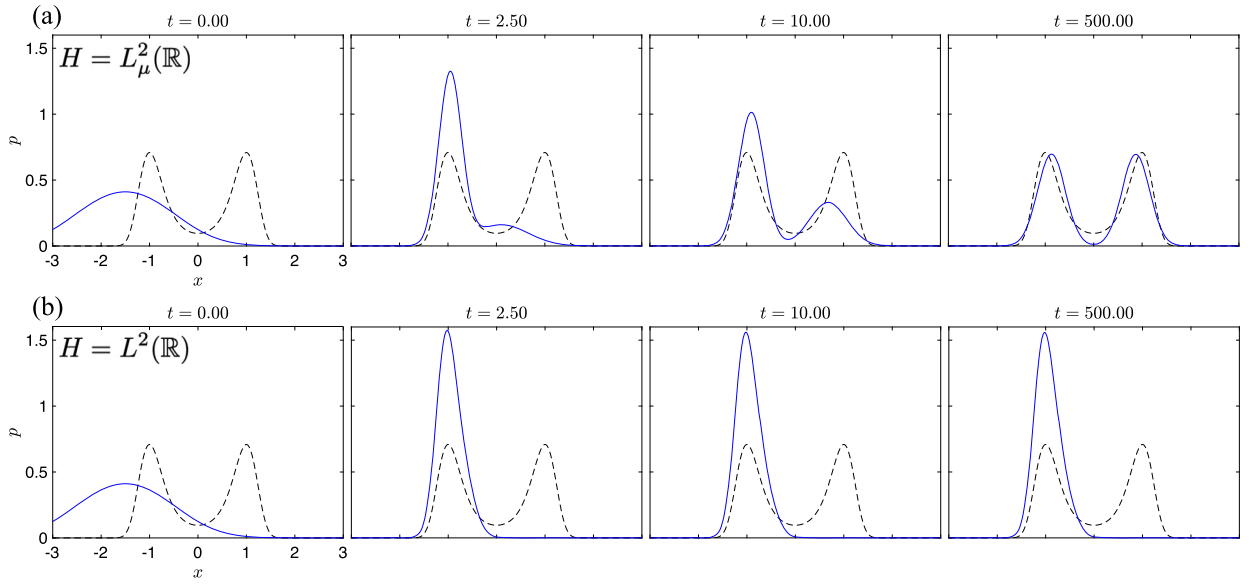

Fig. 3. Evolution of $\hat{p}(x, \boldsymbol{\theta}(t))$ from applying RONS using 2 Gaussians in the approximate solution (blue curves). The true equilibrium density $p_{\text{eq}}(x)$ is marked by the dashed black curve. The initial condition is $A_i(0) = 1/2$, $L_i(0) = 2/\sqrt{\pi}$, $c_1(0) = -1$, $c_2(0) = -2$. The Hilbert space is (a) $H = L^2_\mu(\mathbb{R})$ and (b) $H = L^2(\mathbb{R})$.

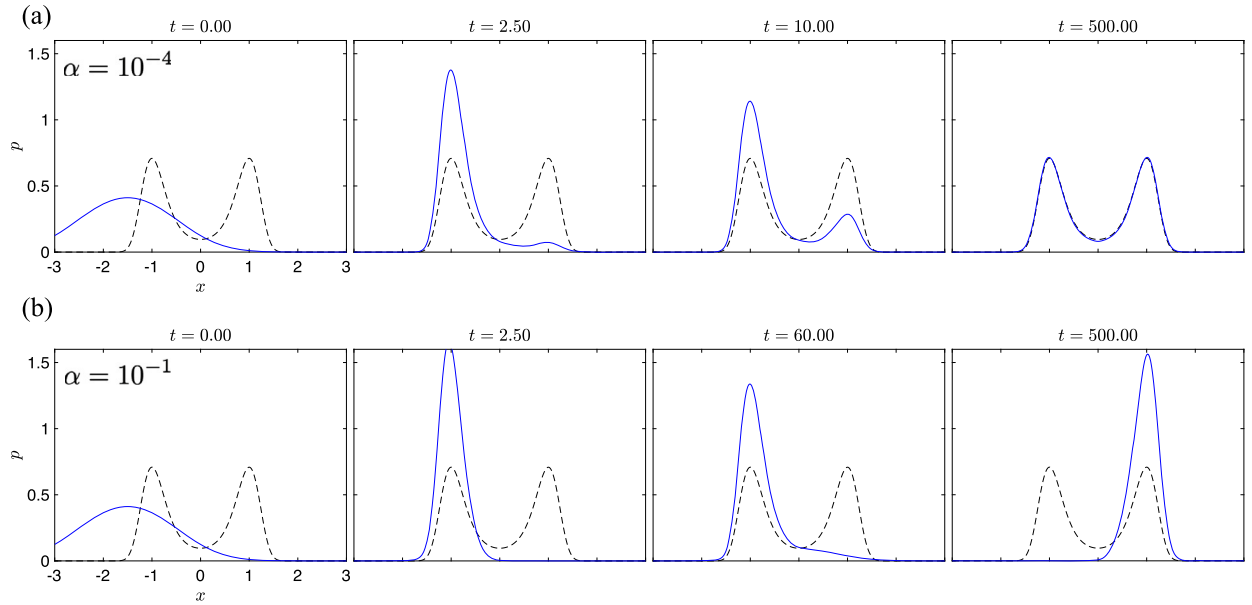

Fig. 4. Evolution of $\hat{p}(x, \boldsymbol{\theta}(t))$ from applying RONS with the Hilbert space $H = L^2(\mathbb{R})$ and using 10 Gaussians in the approximate solution (blue curves). We also mark the true equilibrium density $p_{\text{eq}}(x)$ with a dashed black curve. The initial condition is $A_i(0) = 1/\sqrt{20}$, $L_i(0) = 2/\sqrt{\pi}$, where the amplitudes are chosen so that total probability is 1. Half of the Gaussians are initially placed at $x = -1$, with the other half placed at $x = -2$. Regularization parameter is (a) $\alpha = 10^{-4}$ and (b) $\alpha = 10^{-1}$.

Fig. 4(b), choosing $\alpha = 10^{-1}$ causes the solution to approach the incorrect equilibrium again. Although here we chose the Tikhonov regularization parameter $\alpha = 10^{-4}$ by trial and error, there exist more rigorous methods for choosing this parameter a priori [18,25].

Fig. 5 shows the equilibrium error when using $r = 10$ Gaussians with both the weighted and unweighted inner products. A comparison of computation time and errors is also provided in Table 1. Although in both cases the errors are small, using the Hilbert space $L^2_\mu(\mathbb{R})$ provides a more accurate solution than the Hilbert space $L^2(\mathbb{R})$. However, as we noted earlier, using the unweighted inner product space $L^2(\mathbb{R})$ is scalable to higher dimensions since it allows the use of symbolic computing instead of collocation points (cf. Section 5.4 below).

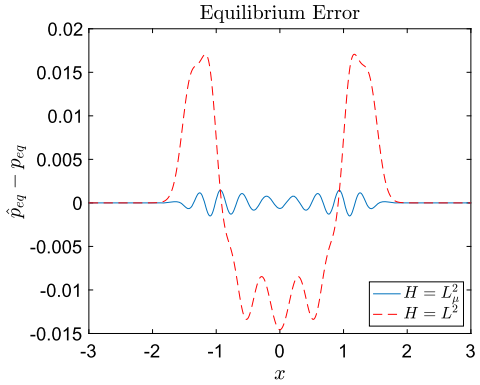

Fig. 5. Error in approximating the true equilibrium density p_{eq} using 10 Gaussians in the approximate solution, regularization parameter $\alpha = 10^{-4}$, and two different choices of Hilbert space. The initial conditions are $A_i(0) = 1/\sqrt{20}$, $L_i(0) = 2/\sqrt{\pi}$.

Table 1
Comparison of computational time and errors for the 1D bistable potential using symbolic RONS (S-RONS) and collocation RONS (C-RONS).

Bistable Potential ($r = 10$)	Symbolic computation	Time integration	Relative error of equilibrium
C-RONS ($H = L^2_\mu(\mathbb{R})$)	none	2.11 seconds	0.002
S-RONS ($H = L^2(\mathbb{R})$)	2.78 seconds	1.50 seconds	0.03

5.3. Stochastic Duffing oscillator

In this section, we consider the stochastic Duffing oscillator [28,39,48] excited by white noise,

$$\mathrm{d}\mathbf{X} = \begin{pmatrix} y \\ a_1 x + a_2 y + a_3 x^3 \end{pmatrix} \mathrm{d}t + \sigma \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \mathrm{d}\mathbf{W}. \tag{36}$$

Here $\mathbf{X} = (x, y)^\top$ where $x(t)$ is the displacement and $y(t) = \dot{x}(t)$ is the velocity. The vector $\mathbf{W}(t) = (W_1(t), W_2(t))^\top$ represents the standard Wiener process in two dimensions. The coefficients a_i are constants where a_1 controls the stiffness of the oscillator, a_2 controls the damping, and a_3 controls the strength of the nonlinearity in the restoring force of the oscillator. The Fokker–Planck equation for the stochastic Duffing oscillator is given by

$$\frac{\partial p}{\partial t} = -\left[y \frac{\partial p}{\partial x} + a_2 p + (a_1 x + a_2 y + a_3 x^3) \frac{\partial p}{\partial y} \right] + \frac{\sigma^2}{2} \frac{\partial^2 p}{\partial y^2}. \tag{37}$$

An analytic solution to the Fokker–Planck equation (37) is not known for all times. However, the asymptotically stable equilibrium solution is given by [39]

$$p_{\text{eq}}(\mathbf{x}) = C \exp\left[\frac{-a_1 a_2 x^2 - \frac{1}{2} a_2 a_3 x^4 + a_2 y^2}{\sigma^2} \right], \tag{38}$$

where C is a normalizing constant. Following [39], we use the parameter values $(a_1, a_2, a_3) = (1, -0.2, -1)$ and a noise intensity of $\sigma = 1/\sqrt{20}$. This leads to a bimodal equilibrium distribution, with peaks at $(x, y) = (\pm 1, 0)$.

To approximate solutions of the Fokker–Planck equation (37), we use the Hilbert space $H = L^2(\mathbb{R}^2)$ and the Gaussian approximate solution (4). We evolve the parameters using the S-RONS equation (11) with the regularization parameter $\alpha = 10^{-3}$. These ODEs are integrated numerically using Matlab's `ode45` [16,43].

Fig. 6 shows the evolution of the approximate solution $\hat{p}(\mathbf{x}, \boldsymbol{\theta}(t))$ with $r = 30$ modes. The initial condition is set up so that 15 Gaussians are centered at $(x, y) = (-1, -1)$ and the other 15 are centered at $(x, y) = (+1, +1)$, leading to a bimodal initial condition. The approximate solution is evolved until it converges to the equilibrium density and virtually no further change is detected.

We compare the S-RONS solution against large-scale Monte Carlo simulations of the Duffing SDE (36). To this end, we evolve 10^6 particles using the predictor-corrector scheme of Ref. [12]. The particles are drawn at random such that their distribution matches that of the S-RONS simulations at the initial time $t = 0$. The evolution of the resulting Monte Carlo PDF is shown in Fig. 7.

Comparing Figs. 6 and 7, we observe that S-RONS, not only returns the correct equilibrium density, but also reproduces the transient dynamics very well. RONS does not perfectly capture some of the finer features seen in the Monte Carlo approach, such as the tails of the solution at $t = 5$. This is expected due to the fact that we are evolving only 30 Gaussians. In fact, increasing the number of terms to $r = 100$ allows us to capture these fine features as well (not shown here for brevity).

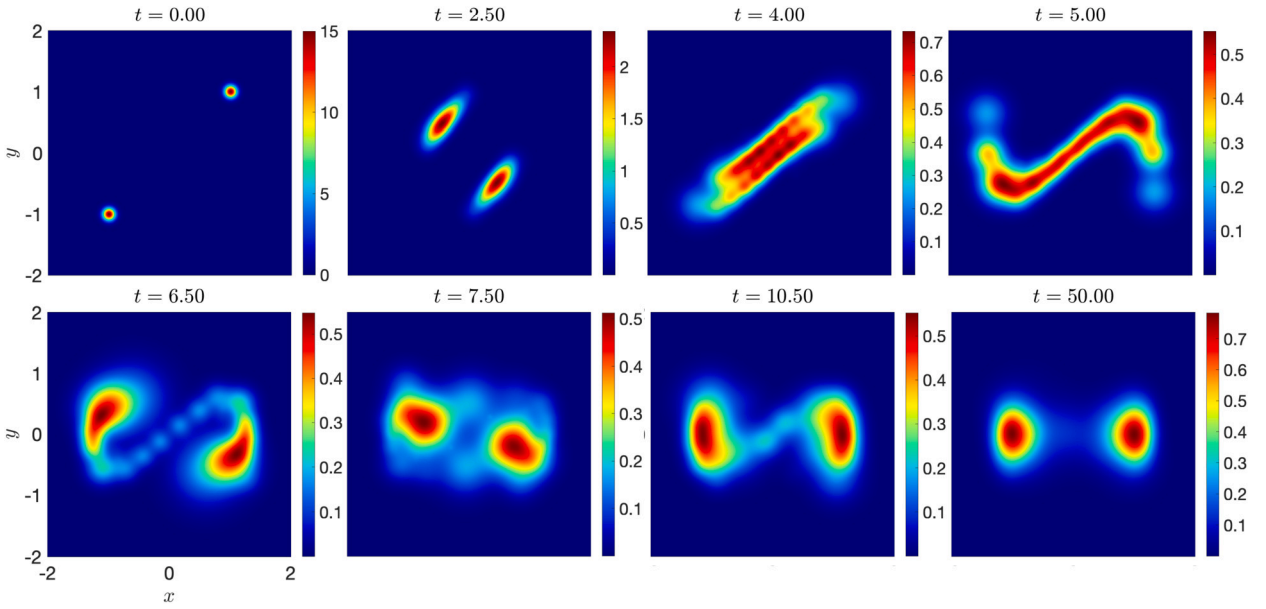

Fig. 6. PDF predicted by applying RONS to the Fokker–Planck equation (37) for the Duffing oscillator excited by white noise. We use 30 Gaussians in the approximate solution. Initial conditions are $A_i(0) = 1$, $L_i(0) = (30\pi)^{-1/2}$, and half the Gaussians are placed at $(x, y) = (-1, -1)$ with the other half at $(x, y) = (1, 1)$.

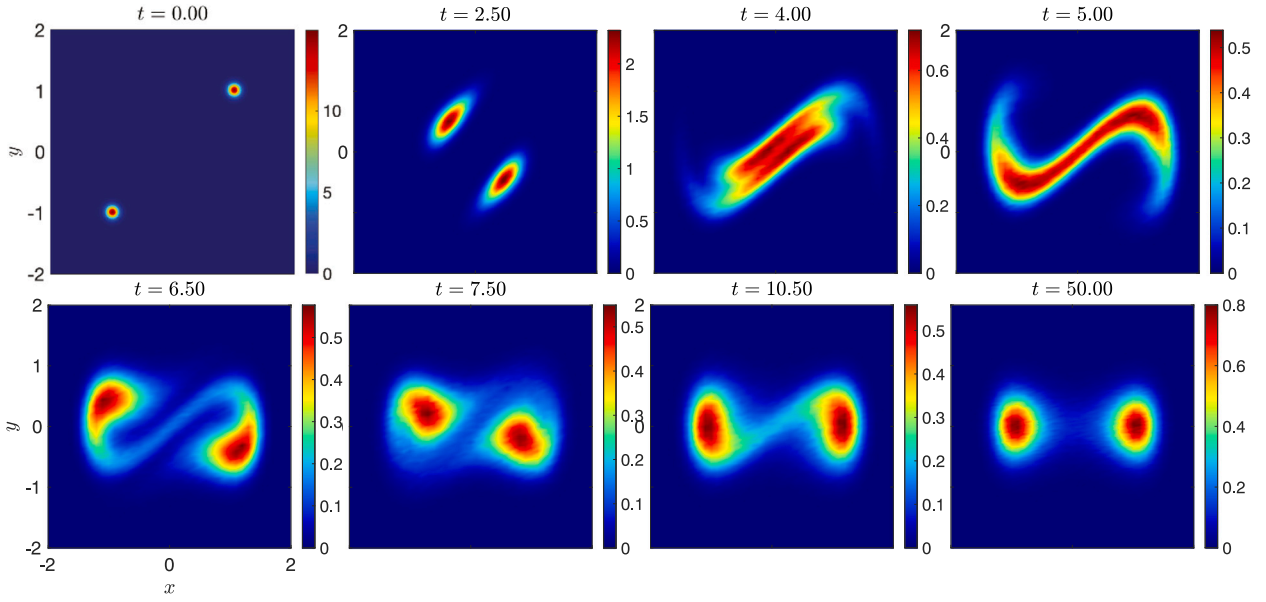

Fig. 7. PDF predicted by direct Monte Carlo simulations of the Duffing oscillator excited by white noise (36). The initial distribution is the sum of 30 Gaussians with parameter values $A_i(0) = 1$, $L_i(0) = (30\pi)^{-1/2}$, and half the Gaussians are placed at $(x, y) = (-1, -1)$ with the other half at $(x, y) = (1, 1)$.

In terms of computational cost, RONS is over 600 times faster than direct Monte Carlo simulations. As reported in Table 2, the Monte Carlo simulations take over 2 hours to complete, whereas RONS takes 3.83 seconds for symbolic computing and approximately 10 seconds for time integration. We emphasize that the symbolic computation for S-RONS only needs to be carried out once; changing the initial condition or increasing the number of modes r does not require additional symbolic computing.

Finally, Fig. 8 compares the RONS solution at time $t = 50$ with the analytical equilibrium density (38). We can see that RONS provides an excellent approximation of the analytical equilibrium position, with an L^2 relative error of 2.2%.

5.4. Harmonic trap

In this section, we study an SDE in eight dimensions driven by a harmonic trap, which was also investigated in [3,9]. More specifically, we consider a system of d interacting particles whose motion is governed by the SDE,

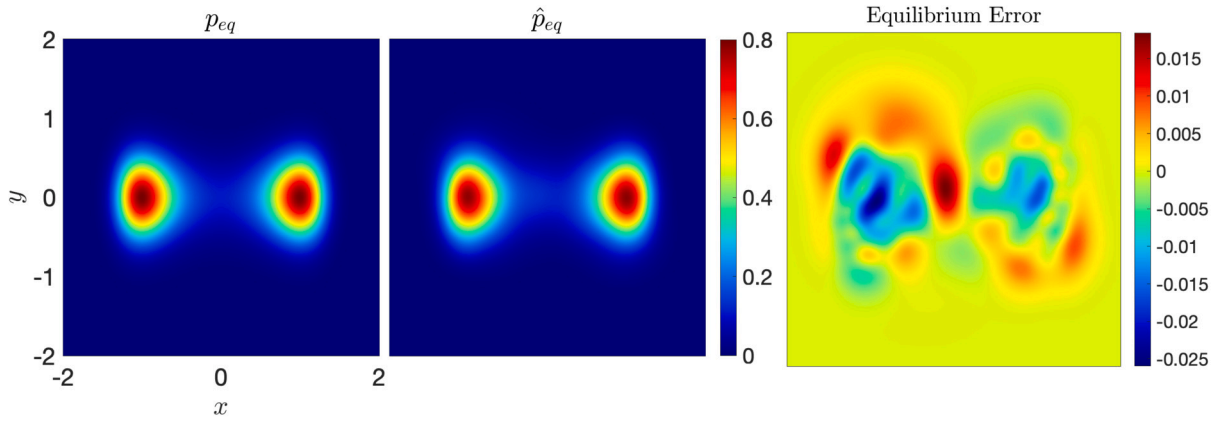

Fig. 8. Comparing the true equilibrium density p_{eq} to the approximate density $\hat{p}_{\text{eq}}$ obtained by RONS using 30 Gaussians. Initial conditions are $A_i(0) = 1$, $L_i(0) = (30\pi)^{-1/2}$, with half the Gaussians placed at $(x, y) = (-1, -1)$ and the other half at $(x, y) = (1, 1)$.

Table 2
Comparison of computation time for stochastic Duffing oscillator example using symbolic RONS and Monte Carlo simulations of the SDE (36).

Stochastic Duffing Oscillator ($r = 30$)	Symbolic computation	Time integration
Monte Carlo Simulations	none	136.44 minutes
Symbolic RONS	3.83 seconds	9.56 seconds

$$\mathrm{d}X_i = g(t, X_i)\mathrm{d}t + \sum_{j=1}^{d} K(X_i, X_j)\mathrm{d}t + \sigma \mathrm{d}W_i, \quad i = 1, 2, ..., d, \tag{39}$$

where $X_i(t)$ denotes the position of the i-th particle. The function $g : [0, \infty] \times \mathbb{R} \to \mathbb{R}$ is a forcing term and $K : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ describes the interaction between particles. The corresponding Fokker–Planck equation is given by

$$\frac{\partial p}{\partial t} = \sum_{i=1}^{d} -\frac{\partial}{\partial x_i}\left[\left(g(t, x_i) + \sum_{j=1}^{d} K(x_i, x_j)\right)p\right] + \nu \frac{\partial^2 p}{\partial x_i^2}, \tag{40}$$

where $\nu = \sigma^2/2$.

As in [9], we set

$$g(t, x_i) = a(t) - x_i, \quad K(x_i, x_j) = \frac{\gamma}{d}(x_j - x_i). \tag{41}$$

The choice of g corresponds to particles in a harmonic trap centered around $a(t)$ while the particles also attract each other due to interaction term K. A significant advantage of these choices for g and K is that we can directly compute the mean and covariance of the particles to serve as a benchmark for our RONS results. By taking the expected value of the SDE (39), we obtain an expression for the mean of each particle

$$\dot{\bar{X}}_i = a(t) - \bar{X}_i + \frac{\gamma}{d}\sum_{j=1}^{d}(\bar{X}_j - \bar{X}_i), \quad i = 1, 2, ..., d, \tag{42}$$

where $\bar{X}_i = \mathbb{E}[X_i]$. We can similarly derive an expression for the evolution of the correlation matrix $\Sigma_{ij} = \mathbb{E}[X_i X_j]$,

$$\dot{\Sigma}_{ij} = a(t)(\bar{X}_j + \bar{X}_i) - 2(1 + \gamma)\Sigma_{ij} + \frac{\gamma}{d}\sum_{l=1}^{d}(\Sigma_{lj} + \Sigma_{li}) + 2\nu\delta_{ij}, \qquad i, j \in \{1, 2, ..., d\}, \tag{43}$$

where δ_{ij} denotes the Kronecker delta. The covariance matrix, whose entries are given by $\Sigma_{ij} - \bar{X}_i\bar{X}_j$, is then calculated using the solutions of (42) and (43).

For the Fokker–Planck equation, we choose the initial condition,

$$p(\mathbf{x}, 0) = \left(2\pi\sigma_0^2\right)^{-d/2} \exp\left[-\frac{|\mathbf{x} - \boldsymbol{\mu}|^2}{2\sigma_0^2}\right], \tag{44}$$

where $\sigma_0^2 = 0.1$ and $\boldsymbol{\mu} \in \mathbb{R}^d$ is given by $\mu_i = i - 1$ for $i = 1, ..., d$. The remaining parameters are given by $a(t) = 1.25(\sin(\pi t) + 1.5)$, $\gamma = 0.25$, $d = 8$, and $\nu = 0.01$.

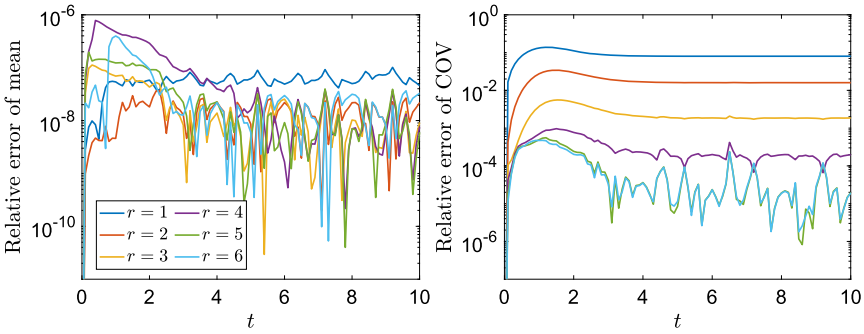

Fig. 9. RONS simulation of the harmonic trap using various numbers of modes with regularization parameter $\alpha = 10^{-8}$.

Table 3
Comparison of computational time and accuracy for harmonic trap example using S-RONS for various numbers of modes r and regularization parameter values α.

Harmonic Trap	Symbolic computation	Time integration	Relative error of mean	Relative error of covariance
$r = 1, \alpha = 10^{-8}$	13.7 minutes	0.06 seconds	$\approx 5 \times 10^{-8}$	$\approx 8 \times 10^{-2}$
$r = 2, \alpha = 10^{-8}$	0 minutes	0.34 seconds	$\approx 10^{-8} - 10^{-9}$	$\approx 2 \times 10^{-2}$
$r = 3, \alpha = 10^{-8}$	0 minutes	1.32 seconds	$\approx 10^{-8} - 10^{-9}$	$\approx 2 \times 10^{-3}$
$r = 4, \alpha = 10^{-8}$	0 minutes	16.44 seconds	$\approx 10^{-8} - 10^{-9}$	$\approx 2 \times 10^{-4}$
$r = 5, \alpha = 10^{-8}$	0 minutes	85.00 seconds	$\approx 10^{-8} - 10^{-9}$	$\approx 10^{-5}$
$r = 6, \alpha = 10^{-8}$	0 minutes	470.50 seconds	$\approx 10^{-8} - 10^{-9}$	$\approx 10^{-5}$
$r = 6, \alpha = 10^{-7}$	0 minutes	3.26 seconds	$\approx 10^{-6} - 10^{-7}$	$\approx 10^{-4}$

We use S-RONS with the Hilbert space $H = L^2(\mathbb{R}^8)$ to evolve the approximate solution (4). For this approximate solution to coincide with the initial condition (44) at time $t = 0$, we choose the parameter values $A_i(0) = (2\pi \times 0.1)^{-4} r^{-1}$, $L_i^2(0) = 0.2$, and $c_i(0) = \boldsymbol{\mu}$. We again enforce the total probability of the approximate solution $\hat{p}(\mathbf{x}, \boldsymbol{\theta})$ to be always equal to one.

We numerically integrate the S-RONS equation (11) with the regularization parameter $\alpha = 10^{-8}$ using Matlab's `ode113s` [43] with a relative and absolute error tolerance of 10^{-8}. Fig. 9 shows the relative error of the mean and covariance when using increasing number of modes r in the approximate solution. Additionally, we report computational times and errors for each simulation in Table 3. We note that the cases with $r = 1$ and $r = 2$ are not stiff and therefore do not require any regularization, but using the same regularization parameter for every run allows for a fair comparison between all simulations. The mean is accurately predicted by RONS, regardless of how many modes we use in the approximate solution. In this example, the true solution is a Gaussian [9], and therefore we expect to capture the true mean within the accuracy of our time integration error tolerances with our approximate solution which is a sum of Gaussians. As a result, the mean is predicted quite accurately even when $r = 1$, and we do not observe a significant improvement in prediction of the mean as we use more modes.

On the other hand, the behavior of the covariance is more complex. The true solution has a diagonal covariance matrix at the initial time. But it develops nonzero entries in the off-diagonal elements after the initial time. The approximate solution $\hat{p}$ is the sum of Gaussians with diagonal covariance matrices, and therefore using more modes leads to significant improvement in the approximation of the covariance matrix. This is demonstrated in Fig. 9, showing that as the number of modes increases from $r = 1$ to $r = 5$ the covariance error decreases monotonically, converging to the tolerance of the numerical time integration. However, increasing the number of modes beyond $r = 5$ does not lead to a significant improvement, indicating that 5 modes are adequate to capture the dynamics.

This example also demonstrates the scalability advantages of symbolic RONS as discussed in Section 3.1. Namely, the number of terms r can be increased without incurring additional symbolic computational cost. That is why in Table 3 the symbolic computational cost is zero for $r \geq 2$. As a result, one can easily increase the number of modes until a satisfactory approximation is achieved.

Clearly, the integration time increases as the number of modes r increase. However, the integration time can be reduced by increasing the regularization parameter α. For example, when using 6 modes and increasing the regularization parameter value to $\alpha = 10^{-7}$ from 10^{-8}, time integration takes only 3.26 seconds. With this regularization parameter, the relative error of the mean is on the order of $10^{-6} - 10^{-7}$ and relative error of the covariance is on the order of 10^{-4} (cf. Table 3).

In the above examples, we assumed that the initial condition p_0 lies on the statistical manifold $\mathcal{M}$ and therefore it can be exactly represented by expansion (4) at time $t = 0$. If the true initial condition p_0 does not initially lie on the statistical manifold, one can solve the optimization problem,

$$\boldsymbol{\theta}_0 = \underset{\boldsymbol{\theta} \in \Omega}{\operatorname{argmin}} \, \| \hat{p}(\cdot, \boldsymbol{\theta}) - p_0 \|_H^2, \tag{45}$$

to find the closest point on the manifold $\mathcal{M}$ to the initial condition p_0. The machine learning community has developed efficient methods, such as stochastic gradient descent [29,42], for solving such optimization problems. These methods are capable of approximating the optimizer even when the number of parameters n is very large. Given that (45) needs to be solved only once at the initial time, this should not present a significant increase in the computational cost of RONS.

6. Conclusions

We showed that the method of reduced-order nonlinear solutions (RONS) leads to a fast and scalable method for approximating the solutions of the Fokker–Planck equation. In particular, we considered the approximate solution as the sum of shape-morphing modes, where the modes are Gaussians with time-dependent amplitudes, means and covariances. RONS equations provide a system of ODEs for optimally evolving the shape parameters such that the approximate solution stays close to a true solution of the PDE. The feasibility of approximating the probability density with a sum of Gaussians is guaranteed by the universal approximation theorem of Park and Sandberg [37].

We demonstrated the efficacy of RONS on several examples. First, we considered the Ornstein–Uhlenbeck process, where the exact solution to the corresponding Fokker–Planck equation is known. In this case, RONS reproduces this exact solution. We also considered three more complex examples, showing that RONS returns accurate approximations of the transient dynamics as well as the equilibrium density. At the same time, RONS is considerably faster than conventional methods. For instance, in the case of the stochastic Duffing oscillator, RONS is 600 times faster than direct Monte Carlo simulations.

We considered two computational methods for forming the RONS equations: symbolic RONS (or S-RONS) and collocation RONS (or C-RONS). In symbolic RONS, we use symbolic computing to evaluate the inner products on the underlying Hilbert space H. This requires only 9 symbolic computation which is independent of the dimension of the system and the number of terms in the approximate solution. If the underlying Hilbert space is chosen to be the Lebesgue space L^2, existing symbolic computing packages easily return closed-form symbolic expressions for the required inner products. We also showed that, if we use the weighted Lebesgue space $H = L^2_\mu$ where $\mu = \hat{p}^{-1}\mathrm{d}\mathbf{x}$, the metric tensor in RONS coincides with the Fisher information matrix defined on statistical manifolds.

Our numerical experiments show that this choice of the underlying Hilbert space ($H = L^2_\mu$) leads to a more accurate approximation of the true solution. However, existing symbolic computing packages did not return a closed-form expression for the required inner products on the space L^2_μ. In such cases, where obtaining symbolic expressions is not feasible, we used C-RONS which minimizes the error at prescribed collocation points. Consequently, C-RONS does not require computing any functional inner products and therefore is applicable in any function space.

Given its higher accuracy, scalability, and connection to Fisher information, using symbolic computing in the Hilbert space L^2_μ is highly desirable. Future work will explore possible avenues for incorporating symbolic computing with this weighted Lebesgue space.

Two open problems remained unanswered in this paper. One concerns the optimal value of the regularization parameter α. Although here we were able to find suitable regularization parameters by trial and error, a systematic method for choosing this parameter is desirable. In our experience, standard methods, such as the L-curve method [11,24], generalized cross-validation (GCV) [15,21,52], and the quasi-optimality criterion [5,4,51], fail to return a suitable regularization parameter for RONS. More specifically, the L-curve method and quasi-optimality criterion significantly over-regularized, which led to inaccurate solutions. On the other hand, GCV significantly under-regularized which caused the time integration to be slow.

The other open problem relates to the number of terms r in the approximation solution (4). Universal approximation theorems guarantee that, for large enough r, the probability density can be approximated with arbitrary accuracy. But they do not provide a rigorous method for choosing r. To reduce the computational cost, one needs to keep r as small as possible, but large enough to guarantee accuracy of the corresponding solution. A systematic method for selecting an optimal r would require error estimates for the shape-morphing solutions of the Fokker–Planck equation. Classical error estimates are not immediately applicable since the shape-morphing approximate solution depends nonlinearly on their parameters. As such, the derivation of error estimates remains an open problem whose resolution will inform the optimal choice of r.

Funding

This work was supported by the National Science Foundation through the award DMS-2208541.

Data availability

No data was used for the research described in the article.

References

[1] W. Anderson, M. Farazmand, Evolution of nonlinear reduced-order solutions for PDEs with conserved quantities, SIAM J. Sci. Comput. 44 (2022) A176–A197, https://doi.org/10.1137/21M1415972.

[2] W. Anderson, M. Farazmand, Shape-morphing reduced-order models for nonlinear Schrödinger equations, Nonlinear Dyn. 108 (2022) 2889–2902, https://doi.org/10.1007/s11071-022-07448-w.

[3] W. Anderson, M. Farazmand, Fast and scalable computation of shape-morphing nonlinear solutions with application to evolutional neural networks, J. Comput. Phys. 498 (2024) 112649, https://doi.org/10.1016/j.jcp.2023.112649.
[4] F. Bauer, M. Reiß, Regularization independent of the noise level: an analysis of quasi-optimality, Inverse Probl. 24 (5) (aug 2008) 055009, https://doi.org/10.1088/0266-5611/24/5/055009.
[5] Frank Bauer, Stefan Kindermann, The quasi-optimality criterion for classical inverse problems, Inverse Probl. 24 (3) (apr 2008) 035002, https://doi.org/10.1088/0266-5611/24/3/035002.
[6] C. Beck, S. Becker, P. Grohs, N. Jaafari, A. Jentzen, Solving the Kolmogorov PDE by means of deep learning, J. Sci. Comput. 88 (2021) 1–28, https://doi.org/10.1007/s10915-021-01590-0.
[7] C. Beck, M. Hutzenthaler, A. Jentzen, B. Kuckuck, An overview on deep learning-based approximation methods for partial differential equations, Discrete Contin. Dyn. Syst., Ser. B 28 (6) (2023) 3697–3746, https://doi.org/10.3934/dcdsb.2022238.
[8] M. Bernstein, L.S. Brown, Supersymmetry and the bistable Fokker-Planck equation, Phys. Rev. Lett. 52 (1984) 1933–1935, https://doi.org/10.1103/PhysRevLett.52.1933.
[9] J. Bruna, B. Peherstorfer, E. Vanden-Eijnden, Neural Galerkin scheme with active learning for high-dimensional evolution equations, Preprint: arXiv:2203.01360, 2022, https://doi.org/10.48550/arXiv.2203.01360.
[10] J. Burbea, C.R. Rao, Entropy differential metric, distance and divergence measures in probability spaces: a unified approach, J. Multivar. Anal. 12 (4) (1982) 575–596, https://doi.org/10.1016/0047-259X(82)90065-3.
[11] D. Calvetti, S. Morigi, L. Reichel, F. Sgallari, Tikhonov regularization and the l-curve for large discrete ill-posed problems, J. Comput. Appl. Math. 123 (1) (2000) 423–446.
[12] W. Cao, Z. Zhang, G. Karniadakis, Numerical methods for stochastic delay differential equations via the Wong–Zakai approximation, SIAM J. Sci. Comput. 37 (1) (2015) A295–A318, https://doi.org/10.1137/130942024.
[13] N. Chen, A.J. Majda, Efficient statistically accurate algorithms for the Fokker–Planck equation in large dimensions, J. Comput. Phys. 354 (2018) 242–268, https://doi.org/10.1016/j.jcp.2017.10.022.
[14] X. Chen, L. Yang, J. Duan, G.E. Karniadakis, Solving inverse stochastic problems from discrete particle observations using the Fokker–Planck equation and physics-informed neural networks, SIAM J. Sci. Comput. 43 (3) (2021) B811–B830, https://doi.org/10.1137/20M1360153.
[15] P. Craven, G. Wahba, Smoothing noisy data with spline functions, Numer. Math. 31 (4) (1978) 377–403.
[16] J.R. Dormand, P.J. Prince, A family of embedded Runge-Kutta formulae, J. Comput. Appl. Math. 6 (1) (1980) 19–26, https://doi.org/10.1016/0771-050X(80)90013-3.
[17] Y. Du, T.A. Zaki, Evolutional deep neural network, Phys. Rev. E 104 (Oct 2021) 045303, https://doi.org/10.1103/PhysRevE.104.045303.
[18] H.W. Engl, M. Hanke, A. Neubauer, Regularization of Inverse Problems, Springer Science & Business Media, vol. 375, The Netherlands, 1996.
[19] M. Farazmand, Mitigation of tipping point transitions by time-delay feedback control, Chaos, Interdiscip. J. Nonlinear Sci. (ISSN 1054-1500) 30 (1) (2020) 01, https://doi.org/10.1063/1.5137825.
[20] R.A. Fisher, On the mathematical foundations of theoretical statistics, Philos. Trans. R. Soc. Lond. Ser. A 222 (594–604) (1922) 309–368, https://doi.org/10.1098/rsta.1922.0009.
[21] G.H. Golub, M. Heath, G. Wahba, Generalized cross-validation as a method for choosing a good ridge parameter, Technometrics (ISSN 0040-1706) 21 (2) (1979) 215–223.
[22] J. Han, A. Jentzen, W. E, Solving high-dimensional partial differential equations using deep learning, Proc. Natl. Acad. Sci. 115 (34) (2018) 8505–8510, https://doi.org/10.1073/pnas.1718942115.
[23] P. Hänggi, P. Jung, Colored noise in dynamical systems, Adv. Chem. Phys. 89 (2007) 239–326.
[24] P.C. Hansen, The l-Curve and Its Use in the Numerical Treatment of Inverse Problems, 1999.
[25] K. Ito, B. Jin, T. Takeuchi, A regularization parameter for nonsmooth Tikhonov regularization, SIAM J. Sci. Comput. 33 (3) (2011) 1415–1438, https://doi.org/10.1137/100790756.
[26] R. Jordan, D. Kinderlehrer, F. Otto, The variational formulation of the Fokker–Planck equation, SIAM J. Math. Anal. 29 (1) (1998) 1–17, https://doi.org/10.1137/S0036141096303359.
[27] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, A. Anandkumar, Neural operator: learning maps between function spaces with applications to PDEs, J. Mach. Learn. Res. 24 (89) (2023) 1–97.
[28] P. Kumar, S. Narayanan, Solution of Fokker–Planck equation by finite element and finite difference methods for nonlinear systems, Sadhana 31 (4) (2006) 445–461, https://doi.org/10.1007/BF02716786.
[29] L. Bottou, Large-scale machine learning with stochastic gradient descent, in: Y. Lechevallier, G. Saporta (Eds.), 19th International Conference on Computational Statistics (COMPSTAT), Heidelberg, Ger.: Physica, 2010, pp. 177–186.
[30] Z. Li, D. Zhengyu Huang, B. Liu, A. Anandkumar, Fourier neural operator with learned deformations for PDEs on general geometries, Preprint: arXiv:2207.05209, 2022, https://doi.org/10.48550/arXiv.2207.05209.
[31] A. Majda, I. Timofeyev, E. Vanden-Eijnden, Stochastic models for selected slow variables in large deterministic systems, Nonlinearity 19 (4) (2006) 769, https://doi.org/10.1088/0951-7715/19/4/001.
[32] K. Mamis, M. Farazmand, Mitigation of rare events in multistable systems driven by correlated noise, Phys. Rev. E 104 (2021) 034201, https://doi.org/10.1103/PhysRevE.104.034201.
[33] A. Mendez, M. Farazmand, Quantifying rare events in spotting: how far do wildfires spread?, Fire Saf. J. 132 (2022) 103630, https://doi.org/10.1016/j.firesaf.2022.103630.
[34] M.K. Murray, J.W. Rice, Differential Geometry and Statistics, Monographs on Statistics and Applied Probability, vol. 48, CRC Press, Washington, D.C., 1993.
[35] M. Naaman, On the tight constant in the multivariate Dvoretzky–Kiefer–Wolfowitz inequality, Stat. Probab. Lett. 173 (2021) 109088, https://doi.org/10.1016/j.spl.2021.109088.
[36] T. Nagler, C. Czado, Evading the curse of dimensionality in nonparametric density estimation with simplified vine copulas, J. Multivar. Anal. 151 (2016) 69–89, https://doi.org/10.1016/j.jmva.2016.07.003.
[37] J. Park, I.W. Sandberg, Universal approximation using radial-basis-function networks, Neural Comput. 3 (2) (1991) 246–257, https://doi.org/10.1162/neco.1991.3.2.246.
[38] L. Pichler, A. Masud, L.A. Bergman, Numerical solution of the Fokker–Planck equation by finite difference and finite element methods—a comparative study, in: Computational Methods in Stochastic Dynamics, vol. 2, 2013, pp. 69–85.
[39] H.J. Pradlwarter, Non-linear stochastic response distributions by local statistical linearization, Int. J. Non-Linear Mech. 36 (7) (2001) 1135–1151, https://doi.org/10.1016/S0020-7462(00)00085-8.
[40] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707, https://doi.org/10.1016/j.jcp.2018.10.045.
[41] C.R. Rao, Information and accuracy attainable in the estimation of statistical parameters, Bull. Calcutta Math. Soc. 37 (3) (1945) 81–91, https://doi.org/10.1007/978-1-4612-0919-5_16.
[42] S. Ray, A quick review of machine learning algorithms, in: 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), 2019, pp. 35–39.

[43] L.F. Shampine, M.W. Reichelt, The Matlab ODE suite, SIAM J. Sci. Comput. 18 (1) (1997) 1–22, https://doi.org/10.1137/S1064827594276424.
[44] B.W. Silverman, Density Estimation for Statistics and Data Analysis, vol. 26, CRC Press, London, 1986.
[45] J. Sirignano, K. Spiliopoulos, DGM: a deep learning algorithm for solving partial differential equations, J. Comput. Phys. 375 (2018) 1339–1364, https://doi.org/10.1016/j.jcp.2018.08.029.
[46] K. Sobczyk, Stochastic Differential Equations, Mathematics and Its Applications, Springer, Dordrecht, 1991.
[47] C. Soize, Steady-state solution of Fokker-Planck equation in higher dimension, Probab. Eng. Mech. (ISSN 0266-8920) 3 (4) (1988) 196–206, https://doi.org/10.1016/0266-8920(88)90012-4.
[48] B.F. Spencer, L.A. Bergman, On the numerical solution of the Fokker–Planck equation for nonlinear stochastic systems, Nonlinear Dyn. 4 (4) (1993) 357–372, https://doi.org/10.1007/BF00120671.
[49] C.J. Stone, Optimal rates of convergence for nonparametric estimators, Ann. Stat. 8 (6) (1980) 1348–1360, https://doi.org/10.1214/aos/1176345206.
[50] K. Tang, X. Wan, Q. Liao, Adaptive deep density approximation for Fokker-Planck equations, J. Comput. Phys. 457 (2022) 111080, https://doi.org/10.1016/j.jcp.2022.111080.
[51] A.N. Tikhonov, V.B. Glasko, Use of the regularization method in non-linear problems, USSR Comput. Math. Math. Phys. (ISSN 0041-5553) 5 (3) (1965) 93–107, https://doi.org/10.1016/0041-5553(65)90150-3.
[52] C.R. Vogel, Computational Methods for Inverse Problems, Society for Industrial and Applied Mathematics, 2002.
[53] G.W. Wei, Discrete singular convolution for the solution of the Fokker–Planck equation, J. Chem. Phys. 110 (18) (1999) 8930–8942, https://doi.org/10.1063/1.478812.
[54] Y. Xu, H. Zhang, Y. Li, K. Zhou, Q. Liu, J. Kurths, Solving Fokker-Planck equation using deep learning, Chaos, Interdiscip. J. Nonlinear Sci. 30 (1) (2020), https://doi.org/10.1063/1.5132840.