Machine Learning: Science and Technology

LETTER • OPEN ACCESS

Variational Monte Carlo approach to partial differential equations with neural networks

IOP Publishing Mach. Learn.: Sci. Technol. **3** (2022) 04LT02 https://doi.org/10.1088/2632-2153/aca317

MACHINE
LEARNING
Science and Technology

CrossMark

OPEN ACCESS

RECEIVED
22 July 2022

REVISED
28 October 2022

ACCEPTED FOR PUBLICATION
15 November 2022

PUBLISHED
1 December 2022

LETTER

Variational Monte Carlo approach to partial differential equations with neural networks

Moritz Reh[1,*] and **Martin Gärttner**[1,2,3,*]

[1] Kirchhoff-Institut für Physik, Universität Heidelberg, Im Neuenheimer Feld 227, 69120 Heidelberg, Germany
[2] Physikalisches Institut, Universität Heidelberg, Im Neuenheimer Feld 226, 69120 Heidelberg, Germany
[3] Institut für Theoretische Physik, Universität Heidelberg, Philosophenweg 16, Heidelberg, 69120, Germany
[*] Authors to whom any correspondence should be addressed.

E-mail: **moritz.reh@kip.uni-heidelberg.de** and **martin.gaerttner@kip.uni-heidelberg.de**

Keywords: normalizing flows, partial differential equations, Fokker–Planck equation, dynamical systems, statistical physics

Supplementary material for this article is available online

Abstract

The accurate numerical solution of partial differential equations (PDEs) is a central task in numerical analysis allowing to model a wide range of natural phenomena by employing specialized solvers depending on the scenario of application. Here, we develop a variational approach for solving PDEs governing the evolution of high dimensional probability distributions. Our approach naturally works on the unbounded continuous domain and encodes the full probability density function through its variational parameters, which are adapted dynamically during the evolution to optimally reflect the dynamics of the density. In contrast to previous works, this dynamical adaptation of the parameters is carried out using an explicit prescription avoiding iterative gradient descent. For the considered benchmark cases we observe excellent agreement with numerical solutions as well as analytical solutions for tasks that are challenging for traditional computational approaches.

1. Introduction

The description of nearly all processes in nature is formalized and modelled by means of differential equations, which dictate the evolution of a system given its initial state. Examples include the Navier–Stokes equation in fluid mechanics [1–4], the Schrödinger equation in quantum mechanics [5–7], and the Fokker–Planck equation governing diffusive processes [8–15]. Analytical solutions of these equations are only available in special cases and, generally, one is forced to resort to numerical techniques. A significant effort during the last century was made to improve the numerical solutions of differential equations [16–18]. There are numerous properties a numerical solver should ideally fulfil, rendering the field quite diverse, with many specialized solvers being developed [19].

Here, we focus on modelling the dynamics of d-dimensional probability density functions (PDFs) by means of an ansatz function, which in our case is given by an artificial neural network (ANN), as illustrated in figure 1. We consider evolution equations of Fokker–Planck form

$$\partial_t p = -\sum_{i}^{d} \partial_{x_i} \mu_i p + \sum_{ij}^{d} \partial_{x_i} \partial_{x_j} D_{ij} p, \tag{1}$$

where $\boldsymbol{\mu} \in \mathbb{R}^d$ is the drift and $D \in \mathbb{R}^{d\times d}$ is the positive semi-definite diffusion matrix and it is understood that p, $\boldsymbol{\mu}$ and D are evaluated at position $\mathbf{x}$ and time t.

PDFs arise naturally across many disciplines, describing, for example, the phase space evolution of (quantum) matter [20, 21], the positions of particles subject to Brownian motion [11], the density of fluids [1] or stock prices in finance [15]. For many of these scenarios the PDF evolution is described by a diffusion process, meaning that the path of a single sampled point evolves according to a stochastic differential

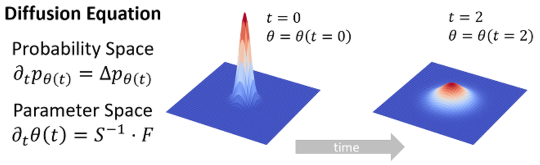

Figure 1. Illustration of the variational approach for a simple diffusion process in 2D. The parameters $\theta(t = 0)$ of the artificial neural network encode a Gaussian at time $t = 0$ and are adapted such that they accurately track the time evolution dictated by the diffusion equation until later times $t = 2$, representing a Gaussian with increased variance.

equation (SDE) [22]. In the limit of averaging infinitely many stochastic trajectories one recovers the evolution of the PDF.

Consequently, the temporal evolution of probability densities can be obtained by either directly solving equation (1) via spatial discretization (grid based solvers), or by solving the corresponding stochastic dynamics for a large number of sample points (particle based solvers). The former approach, while allowing to control the discretization error via the grid spacing, suffers from the curse of dimensionality [23, 24] as the computational cost scales exponentially in the spatial dimension, restricting its applicability to low dimensional cases. The latter approach solves the SDE associated to the Fokker–Planck equation through the Feynman–Kac formula for an ensemble of points sampled from the initial distribution [25, 26]. While suited to compute observables, such as moments of the distribution, in high dimensions, there is no direct way to obtain estimates for functionals of the distribution as an expression for p is lacking [27–29].

In this work, we present a new tool that overcomes the aforementioned limitations of traditional methods by combining variational Monte-Carlo (VMC) with normalizing flows (NFs). While VMC is a long established technique in quantum many-body physics [30–33], NFs are a relatively novel class of ANNs also known as invertible neural networks (INNs) [34]. They have been applied with remarkable success to long standing problems in statistical physics [35], inference and data generation [34, 36–40], as well as quantum field theories [41, 42]. Here, we understand the NF as an ansatz function for the time-dependent density. The choice of the ansatz-function is a degree of freedom in our approach and can be adapted to the problem at hand exploiting prior knowledge about the function class the time-dependent density belongs to. Among the possible choices, ANNs are a promising class of ansatz functions, as they may become *universal* function approximators in the infinite parameter limit, which applies to lesser extent to NFs [43, 44]. Adjusting the parameters of the ansatz function to the dynamics dictated by equation (1) is achieved by a time-dependent variational principle (TDVP), which maps the dynamics of the PDF onto the variational manifold generated by the ansatz function [32, 33, 45]. Crucially, the approach is self-contained and at no point relies on data generated from other solvers, in contrast to prior works using neural networks to solve partial differential equations (PDEs) [46–49], allowing us to obtain numerical solutions for tasks that are challenging for grid-based or particle-based solvers. Our approach differs from the popular physics informed neural networks (PINNs) [50, 51] in that we do not carry out a costly global gradient-descent based optimization in each time step to update the models' parameters, but rather follow an explicit, analytically derived time derivative of the network parameters which is given by the TDVP. We are particularly interested in high-dimensional scenarios which are infeasible to solve with grid-based methods and in quantities which are not easily obtainable by modelling many stochastic processes, such as functionals of the PDF. Indeed we show that, using the developed approach, we can reliably estimate differential entropies in a Monte Carlo fashion requiring only a few thousand samples. We benchmark our approach for the case of an eight-dimensional heat equation and a six-dimensional dissipative phase space evolution.

2. Normalizing flows

While we employ neural networks as ansatz functions, we emphasize that the derived TDVP is applicable to any parameterized density, such as Gaussian mixture models or energy-based estimators. We use NFs [34, 36] to model densities as they have many desirable properties, among which are (a) a guarantee of normalization for any set of parameters θ, (b) a tractable likelihood and (c) the ability to generate independent samples without the need to resort to Markov Chains. NFs parameterize densities by assuming a latent distribution π which is transformed into the distribution of interest by a trainable and invertible map $\mathbf{f}_\theta$,

$$\mathbf{x} = \mathbf{f}_\theta(\mathbf{z})\text{with } \mathbf{z} \sim \pi. \tag{2}$$

Usually, π is chosen to be a 'simple' distribution, e.g. a Gaussian, such that its samples $\mathbf{z}$ can be generated easily. The probability associated with the point $\mathbf{x}$ is proportional to $\pi(\mathbf{f}_\theta^{-1}(\mathbf{x}))$ times the determinant of the Jacobian of the transformation,

$$p_\theta(\mathbf{x}) = \pi(\mathbf{f}_\theta^{-1}(\mathbf{x})) \left| \det \left(\frac{\partial \mathbf{f}_\theta^{-1}(\mathbf{x})}{\partial \mathbf{x}} \right) \right| . \tag{3}$$

The function $\mathbf{f}_\theta$ is composed from a series of invertible transformations $\mathbf{f}_\theta = \varphi_\theta^1 \circ .. \circ \varphi_\theta^N$ which are explained in detail in the supplemental material (SM). Importantly, the Jacobian of the function is tractable meaning that its determinant is efficiently inferred when computing a forward pass, an operation carried out whenever the real space probability is evaluated at some point of interest. By stacking many of these 'coupling blocks' φ^i, the function $\mathbf{f}_\theta$ becomes an expressive coordinate transform, that is, however, incapable of changing the tail behaviour of the latent space distribution [52]. We overcome this problem by dynamically adapting the latent space distribution π to reflect dynamical changes in the tails of the distribution. This is explained in more detail below and in the SM.

3. Time-dependent variational principle

The idea of the TDVP originated in the context of VMC [30] where it has been applied extensively to solve problems in quantum-many-body physics, with a growing interest in the use of neural networks as variational ansatz functions [32, 33, 45, 53]. Its aim is to locally search for the closest approximation to the dynamics of the density within the variational manifold. Concretely, one aims to solve

$$\underset{\dot{\theta}}{\operatorname{argmin}}\, \mathcal{D}(p_{\theta(t)+\tau\dot{\theta}}, p_{\theta(t)} + \tau \dot{p}_{\theta(t)}) \tag{4}$$

where $\mathcal{D}$ is a suitable distance measure between probability distributions, τ denotes a small time step, $\dot{p}_{\theta(t)}$ is the derivative given by equation (1), and $\dot{\theta}$ is the unknown corresponding parameter time derivative. The solution to equation (4) can be found by requiring the derivative with respect to $\dot{\theta}$ to be zero. By expanding equation (4) to second order in τ one finds

$$S_{kk'}\dot{\theta}_{k'} = F_k. \tag{5}$$

We defer the details of this derivation to the SM. Here $S_{kk'} = \langle O_k(\mathbf{x}) O_{k'}(\mathbf{x}) \rangle_{\mathbf{x}\sim p_\theta(t)}$ denotes the Fisher information metric and $F_k = \langle O_k(\mathbf{x}) \partial_t \log(p_{\theta(t)}(\mathbf{x})) \rangle_{\mathbf{x}\sim p_{\theta(t)}}$ is a force term, where O_k denotes the (logarithmic) variational derivative $O_k(\mathbf{x}) = \partial_{\theta_k} \log(p_{\theta(t)}(\mathbf{x}))$and $\partial_t \log(p_{\theta(t)}(\mathbf{x}))$ is given by the RHS of the to be solved PDE. Here $\langle \cdot \rangle_{\mathbf{x}\sim p_\theta(t)}$ denotes an expectation value evaluated through Monte Carlo sampling from the model distribution $p_\theta(t)$. Notice, that we heavily rely on the differentiability of the ansatz function $p_{\theta(t)}$ with respect to both variational parameters and spatial coordinates. The latter frequently appear on the RHS of equation (1) and are thus required for computing $\dot{p}_{\theta(t)}$. This is in striking contrast to grid-based techniques which require making grid cells finer for higher accuracy. Here, instead, we have access to the exact derivatives through automatic differentiation. The choice of distance measure to compare the two probability distributions is not arbitrary as the form of S and F directly depends on it. In order to obtain expressions of S and F that can be efficiently estimated through a finite number of samples, we found that both the Hellinger distance $\mathcal{D}_\mathrm{H}(p,q) = 1 - F(p,q) = 1 - \int \sqrt{pq}d\mathbf{x}$ and the Kullback–Leibler (KL) divergence $\mathcal{D}_\mathrm{KL}(p,q) = \int p \log(p/q) d\mathbf{x}$ yield the same result of the desired form. Care has to be taken when solving equation (5) for $\dot{\theta}$, as the inverse of S may not exist. This is the case if directions in parameter space are present along which the probabilities are stationary, which can be dealt with by regularization procedures [33, 53].

4. Problem setup

We are interested in solving initial value problems, for which the initial density distribution $p(0,\mathbf{x}) = u(\mathbf{x})$ is given along with the RHS of equation (1) which governs its evolution. To exactly encode the initial distribution $u(\mathbf{x})$ in the model $p_{\theta(t=0)}$, the latent distribution is set to $u(\mathbf{x})$ and the parameters of the map $\mathbf{f}_{\theta(t=0)}$ are chosen such that it represents the identity map $\mathbf{f}_{\theta(t=0)}(\mathbf{x}) = \mathbf{x}$. If the initial distribution cannot be given in closed form and therefore cannot be set analytically as the latent space distribution π, the network may be trained on its samples to approximately encode it at time $t = 0$. Then a solver is used which integrates the parameters according to equation (5).

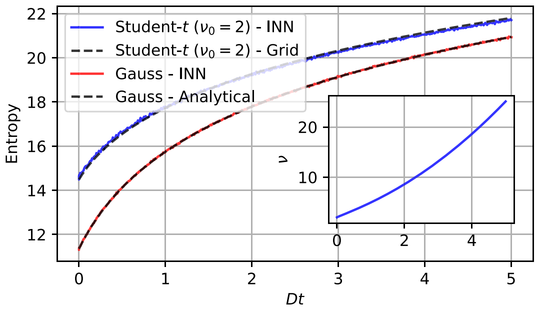

Figure 2. Evolution of the differential entropy under a heat equation for different initial distributions. While analytical comparison data is available in the case of a Gaussian initial distribution, we compare to numerical data for the Student-t obtained with finite differences on a 1D Grid. Inset: adjustment of the latent space distribution π by changing its parameter ν dynamically in time.

5. Application 1: diffusion in high dimensions

As a first benchmark scenario we consider the heat equation in $d = 8$ dimensions. The heat equation appears across many disciplines ranging from engineering [54, 55] and molecular motion [11] to the pricing of financial derivatives given by the famous Black–Scholes equation [56, 57] and reads

$$\partial_t p(t,\mathbf{x}) = D\Delta_{\mathbf{x}} p(t,\mathbf{x}). \tag{6}$$

Importantly, an analytical solution exists against which we can benchmark, making the described scenario a good showcase of the proposed approach. The solution is given by a convolution of the initial distribution $p(0,\mathbf{x})$ with the 'heat kernel' $\Phi(t,\mathbf{x}) = (4\pi t)^{-(d/2)} \exp(-\mathbf{x}^2/4Dt)$ [58], which is the Green's function to equation (6), such that

$$p(t,\mathbf{x}) = \int p(0,\mathbf{y})\Phi(t,\mathbf{x}-\mathbf{y})d\mathbf{y}. \tag{7}$$

We aim to observe the growth of the differential entropy

$$S(t) = -\int p(t,\mathbf{x})\log(p(t,\mathbf{x}))d\mathbf{x} = -\langle \log(p(t,\mathbf{x}))\rangle_{\mathbf{x}\sim p(t,\mathbf{x})} \tag{8}$$

with time, a task, which is challenging or even intractable using other numerical techniques in high dimensions for the reasons mentioned above [23, 24, 27–29]. In the case of a Gaussian distribution for $p(0,\mathbf{x})$ with zero mean and unit covariance matrix, we obtain a Gaussian of larger variance at later points in time, in which case we observe perfect agreement between the analytical solution and the one obtained using the INN as shown in figure 2. If we choose a Student-t distribution as initial distribution, i.e.

$$p(0,\mathbf{x}) \propto \left(1 + \frac{\mathbf{x}^2}{\nu}\right)^{-(\nu+d)/2} \tag{9}$$

with $\nu = 2$ we can no longer compare to the analytical solution as the involved integrals become infeasible to solve. However, by exploiting the spherical symmetry of the problem, we can map the evolution to an effective 1D problem of the radial dependency of p which we can approximately solve on a grid using finite differences. The grid based solution and that obtained using the INN are generally in good agreement. We observe a slight difference which we attribute to technical challenges of the grid-based approach, which we discuss more elaborately in the SM.

6. Application 2: diffusion in classical phase space

As a second demonstration of the proposed approach we consider classical Hamiltonian dynamics in phase space with additional diffusion. Concretely, we choose the Hamiltonian H to represent coupled harmonic oscillators (coupling strength k) which are in contact with heat baths of different temperatures T_i, such that the solution does not factorize in the eigenbasis of H. We provide the Hamiltonian and its generated phase space flow in the SM. The heat baths lead to diffusion in phatse space, which implies that sampled points of the distribution evolve according to an SDE. We show that the INN faithfully estimates moments of the

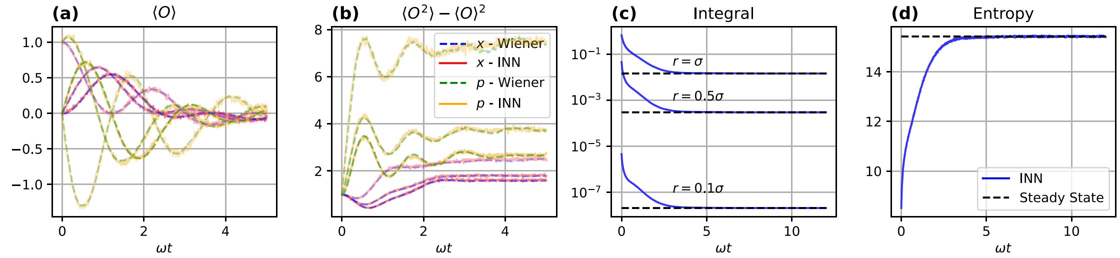

Figure 3. (a) and (b) Evolution of the first two moments of the phase space distribution ρ estimated from 10.000 samples for three coupled harmonic oscillators with dissipation given by the temperatures $k_BT/m\omega^2 = (10,3,1)$ and all other parameters chosen to be unity. The initial distribution in phase space is a Gaussian with unit variance centered at the position $\mathbf{x} = (1,0,0)^T$ and momentum $\mathbf{p} = (0,1,0)^T$. (c) and (d) Three uncoupled oscillators ($k = 0$) coupled to the same heat bath at temperature $k_BT/m\omega^2 = 10$. In (c) the value of the six-dimensional integral around a hypersphere with radius r centered at the origin is shown, while in (d) the estimation of the differential entropy equation (8) using the INN is displayed. Both are shown to converge to the expected value of the steady state. The initial distribution is a Gaussian with identity covariance matrix centered at $\mathbf{x} = (1,1,1)^T$ and zero momentum.

distribution, probabilities (i.e. integrals over finite domains) as well as functionals of the PDF that correspond to integrals over the entire domain.

The described system obeys the following Fokker–Planck equation [59]

$$\begin{aligned}\partial_t\rho(t,\mathbf{x},\mathbf{p}) = \Big[&-\partial_{\mathbf{p}}H\cdot\partial_{\mathbf{x}} + \partial_{\mathbf{x}}H\cdot\partial_{\mathbf{p}} + \\ &\gamma\left(\mathbf{p}\cdot\partial_{\mathbf{p}} + mk_B\sum\nolimits_i T_i\partial^2_{p_i}\right)\Big]\,\rho(t,\mathbf{x},\mathbf{p}),\end{aligned}\tag{10}$$

whose corresponding SDE is given by [59]

$$\begin{aligned}dx_i &= \partial_{p_i}H dt,\\ dp_i &= -\left[\gamma p_i + \partial_{x_i}H\right]dt + \sqrt{\lambda_i}dw_i.\end{aligned}\tag{11}$$

Here, $\lambda_i = \sqrt{2m\gamma k_B T_i}$, $dw_i = \Pi_i\sqrt{dt}$ is the Wiener process with zero average $\langle dw_i\rangle = 0$ and standard scaling $\langle dw_i^2\rangle = dt$ implying that Π is drawn from a standard Gaussian $\Pi \sim \mathcal{N}(0,1)$. For simplicity we choose all quantities except T_i equal to unity.

In the case of heat baths of equal temperatures $T_i = T$ and vanishing coupling ($k = 0$) the system assumes a thermal steady state of Gaussian form in the long time limit given by the Gibbs-ensemble

$$\begin{aligned}\rho_{SS} &= \exp(-H/k_BT)/Z\\ &= \exp\left(-\frac{1}{2}(m\omega^2\mathbf{x}^2 + \mathbf{p}^2/m)/k_BT\right)/Z,\end{aligned}\tag{12}$$

with $Z = \int\exp(-H/k_BT)d\mathbf{x}d\mathbf{p}$ the partition function, where the Gaussian form allows to compare against analytical results.

We consider four quantities of interest which we evaluate by drawing 10.000 samples from the INN, see figure 3. The first two quantities are the means and variances of the distribution evolved for the case of different T_i and $k = 1$. Here, comparison against estimates from solving the SDE for the same number of sampled points is straight forward and one observes excellent agreement between both methods. To obtain an easy benchmark case for integral and entropy estimation, we choose $k = 0$ and $T_i = T$ such that the steady state is Gaussian, see equation (12). We choose the integration volumes to be hyperspheres of radius r centered at the origin allowing for analytical evaluation of the Gaussian integral. The values of these integrals correspond to the probability of finding the system inside the hypersphere. Using the INN, we can estimate such integrals in a Monte-Carlo fashion by uniformly sampling points $\mathbf{x}_i$ from inside the integration domain and average the associated probabilities $p_\theta(\mathbf{x}_i)$, which are shown to converge to the analytically obtained steady-state value in figure 2(c).

Finally, we again focus on the differential entropy (equation (8)), where figure 3(d) shows that our method succeeds to predict the differential entropy with low noise while converging to the expected steady state value.

7. Conclusion and outlook

We have introduced a variational approach to the dynamics of continuous probability distributions using NFs and demonstrated its power by applying it to paradigmatic benchmark problems. Our method is widely applicable, even beyond the Fokker–Planck form (1), e.g. to cases with non-local terms [60]. Its unique strength lies in estimating functionals of probability densities in high dimensions enabled by the availability of exact samples with tractable likelihood. We emphasize that other approaches such as PINN [50] require solving a large-scale non-convex optimization problem in each time step, which the TDVP replaces by the explicit update rule (5) (see SM for further discussion). The form of the ansatz function can be chosen flexibly and is not required to be a neural network. The only restrictions are that (a) samples from its distribution may be obtained and (b) derivatives with respect to inputs and parameters are computable. While building NFs using stacked coupling blocks is a popular approach, other flow architectures exist and it would be interesting to investigate their potential in solving PDEs in the future. Since the TDVP can also work with non-normalized probabilities, also energy based models would be viable ansatz functions although this would mean that samples would have to be obtained by resorting to Markov-chains.

For the utilized architecture we found that challenges exist when trying to solve chaotic dynamics. We believe this to be caused by the high amount of information of the phase space distribution which needs to be encoded using comparably few parameters. Additionally, we found it challenging to model distributions whose tail behaviour deviated from that of the latent space distribution. In the example shown in figure 2 this could be dealt with by elevating ν to be a variational parameter, which would tend to infinity for late times, representing the exact tail behaviour of the real space distribution. However, if the real space tail behaviour cannot be accurately modelled in latent space, e.g. because its form is not known beforehand, one cannot expect to accurately model the distribution on the entire domain.

Data availability statement

The code used for this project is based on the jVMC library [61], making use of flax [62] and jax [63] and is available under GitHub: RehMoritz/vmc_pde. The repository also contains the data from figures 2 and 3.

Acknowledgments

The authors thank Markus Schmitt, Julian Urban, Ullrich Köthe and Peter Sorrenson for helpful discussions. This work is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC2181/1-39 090 0948 (the Heidelberg STRUCTURES Excellence Cluster) and within the Collaborative Research Center SFB1225 (ISOQUANT). This work was partially financed by the Baden-Württemberg Stiftung gGmbH. The authors acknowledge support by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through Grant No. INST 40/575-1 FUGG (JUSTUS 2 cluster). The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS [64] at Jülich Supercomputing Centre (JSC).

ORCID iDs

Moritz Reh https://orcid.org/0000-0002-8408-7558
Martin Gärttner https://orcid.org/0000-0003-1914-7099

References

[1] Ferziger J H and Perić M 2002 *Computational Methods for Fluid Dynamics* (Berlin: Springer)
[2] Tropea C, Yarin A and Foss J 2007 *Springer Handbook of Experimental Fluid Mechanics* (Berlin: Springer)
[3] Wesseling P 2001 *Principles of Computational Fluid Dynamics* (Berlin: Springer)
[4] Spurk J H and Aksel N 2020 *Fluid Mechanics* (Cham: Springer)
[5] Sakurai J J and Napolitano J 2017 *Modern Quantum Mechanics* (Cambridge: Cambridge University Press)
[6] Griffiths D J and Schroeter D F 2018 *Introduction to Quantum Mechanics* (Cambridge: Cambridge University Press)
[7] Schwabl F 2008 *Advanced Quantum Mechanics* (Berlin: Springer)
[8] Kampen N V 2007 *Stochastic Processes in Physics and Chemistry* (Amsterdam: Elsevier)
[9] Coffey W T and Kalmykov Y P 2011 *The Langevin Equation* (Singapore: World Scientific)
[10] Sornette D 2001 *Physica* A **290** 211
[11] Freedman D 1983 *Brownian Motion and Diffusion* (New York: Springer)
[12] Shen X, Chen H, Dai J and Dai W 2002 *Queueing Syst.* **42** 33

[13] Rouse P E 1953 *J. Chem. Phys.* **21** 1272
[14] Prakash J R and Öttinger H C 1999 *Macromolecules* **32** 2028
[15] Reisinger C 2004 *Numerische Methoden für Hochdimensionale Parabolische Gleichungen am Beispiel von Optionspreisaufgaben* (Heidelberg: Heidelberg University Library)
[16] Thomas J W 1995 *Numerical Partial Differential Equations: Finite Difference Methods* (New York: Springer)
[17] Hairer E 2006 *Geometric Numerical Integration* (Berlin: Springer)
[18] Kress R 1998 *Graduate Texts in Mathematics* (New York: Springer) pp 189–224
[19] Quarteroni A and Valli A 1994 *Numerical Approximation of Partial Differential Equations* (Berlin: Springer)
[20] Schwabl F 2006 *Statistical Mechanics* (Berlin: Springer)
[21] Zachos C K, Fairlie D B and Curtright T L 2005 *Quantum Mechanics in Phase Space* (Singapore: World Scientific)
[22] Tomé T and de Oliveira M J 2015 *Stochastic Dynamics and Irreversibility* (Cham: Springer International Publishing)
[23] Kang W and Wilcox L C 2017 *Comput. Optim. Appl.* **68** 289
[24] Griebel M 2006 *Sparse Grids and Related Approximation Schemes for Higher Dimensional Problems* (Cambridge: Cambridge University Press) pp 106–61
[25] Moral P D 2004 *Feynman-Kac Formulae* (New York: Springer)
[26] Oksendal B 1998 *Stochastic Differential Equations* (Berlin: Springer)
[27] Kraskov A, Stögbauer H and Grassberger P 2004 *Phys. Rev.* E **69** 066138
[28] Singh S and Pòczos B 2016 Finite-sample analysis of fixed-k nearest neighbor density functional estimators (arXiv:1606.01554 [math.ST])
[29] Ao Z and Li J 2022 Entropy estimation via normalizing flow (available at: https://aaai-2022.virtualchair.net/poster_aaai8184)
[30] McMillan W L 1965 *Phys. Rev.* **138** A442
[31] Ceperley D, Chester G V and Kalos M H 1977 *Phys. Rev.* B **16** 3081
[32] Carleo G, Cevolani L, Sanchez-Palencia L and Holzmann M 2017 *Phys. Rev.* X **7** 031026
[33] Carleo G and Troyer M 2017 *Science* **355** 602
[34] Dinh L, Sohl-Dickstein J, and Bengio S 2016 Density estimation using Real NVP
[35] Noé F, Olsson S, Köhler J and Wu H 2019 *Science* **365** eaaw1147
[36] Papamakarios G, Nalisnick E, Rezende D J, Mohamed S and Lakshminarayanan B 2019 Normalizing flows for probabilistic modeling and inference (arXiv:1912.02762 [stat.ML])
[37] Grathwohl W, Chen R T Q, Bettencourt J, Sutskever I, and Duvenaud D 2018 Ffjord: free-form continuous dynamics for scalable reversible generative models (arXiv:1810.01367 [cs.LG])
[38] Dinh L, Krueger D and Bengio Y 2014 Nice: non-linear independent components estimation (arXiv:1410.8516 [cs.LG])
[39] Kingma D P and Dhariwal P 2018 Glow: generative flow with invertible 1x1 convolutions (arXiv:1807.03039 [stat.ML])
[40] Ardizzone L, Kruse J, Wirkert S, Rahner D, Pellegrini E W, Klessen R S, Maier-Hein L, Rother C and Köthe U 2018 Analyzing inverse problems with invertible neural networks (arXiv:1808.04730 [cs.LG])
[41] Albergo M S, Kanwar G, Racaniére S, Rezende D J, Urban J M, Boyda D, Cranmer K, Hackett D C and Shanahan P E 2021 *Phys. Rev.* D **104** 114507
[42] Pawlowski J M and Urban J M 2022 Flow-based density of states for complex actions (https://doi.org/10.48550/arXiv.2203.01243)
[43] Kong Z and Chaudhuri K 2020 The expressive power of a class of normalizing flow models (arXiv:2006.00392 [cs.LG])
[44] Teshima T, Ishikawa I, Tojo K, Oono K, Ikeda M and Sugiyama M 2020 Coupling-based invertible neural networks are universal diffeomorphism approximators (arXiv:2006.11469 [cs.LG])
[45] Reh M, Schmitt M and Gärttner M 2021 *Phys. Rev. Lett.* **127** 230501
[46] Brandstetter J, Worrall D and Welling M 2022 Message passing neural PDE solvers (arXiv:2202.03376 [cs.LG])
[47] Lu L, Jin P, Pang G, Zhang Z and Karniadakis G E 2021 *Nat. Mach. Intell.* **3** 218–29
[48] Li Z, Kovachki N, Azizzadenesheli K, Liu B, Bhattacharya K, Stuart A and Anandkumar A 2020 Fourier neural operator for parametric partial differential equations (arXiv:2010.08895 [cs.LG])
[49] Beck C, Becker S, Grohs P, Jaafari N and Jentzen A 2021 *J. Sci. Comput.* **88** 73
[50] Raissi M, Perdikaris P and Karniadakis G 2019 *J. Comput. Phys.* **378** 686
[51] Feng X, Zeng L and Zhou T 2022 *SSRN Electron. J.* **32** 401–23
[52] Jaini P, Selby K A, and Yu Y 2019 Sum-of-squares polynomial flow (arXiv:1905.02325 [cs.LG])
[53] Schmitt M and Heyl M 2020 *Phys. Rev. Lett.* **125** 100503
[54] Poirier D R and Geiger G H 2016 *Transport Phenomena in Materials Processing* (Cham: Springer)
[55] Annaratone D 2010 *Engineering Heat Transfer* (Berlin: Springer)
[56] Black F and Scholes M 1973 *J. Political Econ.* **81** 637
[57] Guillaume T 2018 *Ann. Oper. Res.* **281** 229
[58] Davies E B 1989 *Heat Kernels and Spectral Theory* (Cambridge: Cambridge University Press)
[59] Presilla C, Onofrio R and Patriarca M 1997 *J. Phys. A: Math. Gen.* **30** 7385
[60] Lin L, Duan J, Wang X and Zhang Y 2021 *Chaos* **31** 051105
[61] Schmitt M and Reh M 2022 *SciPost Physics Codebases* p 2
[62] Heek J, Levskaya A, Oliver A, Ritter M, Rondepierre B, Steiner A and vanZee M 2020 Flax: a neural network library and ecosystem forJAX (available at: http://github.com/google/flax)
[63] Bradbury J, Frostig R, Hawkins P, Johnson M J, Leary C, Maclaurin D, Necula G, Paszke A, VanderPlas J, Wanderman-Milne S and Zhang Q 2018 JAX: composable transformations ofPython+NumPy programs (available at: http://github.com/google/jax)
[64] Krause D 2019 JUWELS: modular tier-0/1 supercomputer at the Jülich supercomputing centre *J. Large-Scale Res. Facilities* **5**