# Schrödingerization-based quantum simulation of partial differential equations and related problems – a continuous-variable perspective [*]

Shi Jin [†1,2,3] and Nana Liu [‡1,2,3,4]

[1]School of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai, China.
[2]Institute of Natural Sciences, Shanghai Jiao Tong University, Shanghai, China.
[3]Ministry of Education, Key Laboratory in Scientific and Engineering Computing, Shanghai Jiao Tong University, Shanghai, China
[4]University of Michigan-Shanghai Jiao Tong University Joint Institute, Shanghai, China

January 16, 2025

*Dedicated to the UN International Year of Quantum Science and Technology (IYQ) 2025*

## Abstract

Quantum computers are designed to obey quantum mechanical principles, thus it is most naturally suited to solving the Schrödinger equation – whose evolution operator is *unitary*. This guarantees that it evolves from one pure quantum state – a complex unit vector, generally high dimensional – to another pure quantum state. For other ordinary and partial differential equations whose evolution operator is not unitary, they can not be directly simulated by quantum computers, via amplitude embedding, without some modifications. In [27, 34, 37], a novel technique was introduced, called *Schrödingerization*, which transforms *any* linear ordinary or partial differential equation into a Schrödinger-type equation with unitary evolution, in one higher dimension, thus making it natural for quantum simulation. This article reviews our recent results on Schrödingerization, and how various problems associated with ODE, PDE and linear algebra can be formulated into unitary dynamics using this and other *dimension-lifting* techniques. Our focus will be on the mathematical formulation, using continuous variables, which makes it suitable not only for qubit-based general purpose quantum computers, but also for continuous-variable (or qumodes) based quantum computers running in continuous time, which can be used to design *analog* quantum simulators for specific problems. This makes it more amenable for nearer term quantum devices. Both linear and (some specific) nonlinear partial differential equations will be studied and we will also pose some open questions.

[*] This article is prepared for the Proceedings of the Nineteenth International Conference on Hyperbolic Problems, based on a plenary lecture by the first author.

[†] shijin-m@sjtu.edu.cn

[‡] nana.liu@quantumlah.org

## Contents

# 1 Introduction

Partial differential equations (PDEs) are at the core of computation for science and engineering. However, they may suffer from the curse-of-dimensionality when the spatial dimension is too high, like in quantum dynamics, kinetic and mean-field theory, or when there is uncertainty in the PDE. Some of these equations may also contain multiple or small spatial and temporal scales. These present bottlenecks for classical computation. Quantum algorithms, on the other hand, due to its potential for polynomial or even exponential speedup over its classical counterparts, offers a new computational paradigm that could potentially overcome some of these classical computational bottlenecks for certain problems. It has thus attracted lots of interest in recent years.

Quantum computers are designed with quantum mechanical principles in mind, and quantum algorithms and their corresponding circuits need to be designed following the natural evolution dictated by quantum mechanics. For pure state quantum evolution without noise, it must follow the linear Schrödinger equation

$$i\hbar\partial_t\psi = \boldsymbol{H}\psi, \tag{1.1}$$

where $\psi$ is the complex-valued wave function, $\hbar$ is the Planck constant, $i = \sqrt{-1}$, $\boldsymbol{H}$ is the quantum Hamiltonian defined by

$$\boldsymbol{H} = -\frac{\hbar^2}{2}\Delta + V(\cdot), \tag{1.2}$$

which is a *Hermitian* operator. The evolution of the Schrödinger equation is given by

$$|\psi(t)\rangle = e^{-\frac{i}{\hbar}\boldsymbol{H}t}|\psi(0)\rangle, \tag{1.3}$$

where $|\psi\rangle$ is the quantum state of $\psi$, which is a unit complex vector. Since $\boldsymbol{H}$ is Hermitian, hence the evolution operator $e^{-\frac{i}{\hbar}\boldsymbol{H}t}$ is *unitary.* The key for quantum algorithms is that the evolution, such as quantum gates used to design quantum circuits, *must be unitary* [44, 51]. Since unitary operators preserve Euclidean norms of vectors, the evolution maps pure quantum states to pure quantum states. For most PDEs or dynamical systems (including ordinary differential equations–ODEs), in particular for dissipative systems, their evolution is not unitary, thus it cannot directly take advantage of quantum simulation.

Another difficulty in quantum simulation of PDEs is the difficulty in treating nonlinearity in quantum simulations, since, after all, quantum mechanics – evolving via the linear Schrödinger equation – is linear!

A novel technique, called *Schrödingerization*, which transforms any linear dynamical system, including ODEs and PDEs, to Schrödinger type PDEs, namely with unitary evolution operators, in one higher dimension, was introduced in [27, 34, 37]. This enables quantum simulation for general linear ODEs and PDEs, in a simple and generic way. In the next section we summarize this approach.

# 2 Schrödingerization

## 2.1 General linear ODEs and PDEs

A general linear systems ODEs or PDEs for $\boldsymbol{u}(t)$ that is first-order in time $t \geq 0$ can be written as

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{A}(t)\boldsymbol{u}, \quad \boldsymbol{u}(t=0) = \boldsymbol{u}_0 \tag{2.1}$$

where $\boldsymbol{A}(t)$ is a (possibly time-dependent, or non-autonomous) linear operator for ODEs and a linear differential operator for PDEs. In general the evolution operator $\mathcal{T}e^{\int_0^t \boldsymbol{A}(\tau)d\tau}$ (where $\mathcal{T}$ is the time-ordering operator, needed since in general $[\boldsymbol{A}(t), \boldsymbol{A}(t')] \neq 0$ for $t \neq t'$, where $[\cdot,\cdot]$ is the commutator [55]) is not unitary thus the system is not directly suitable for quantum simulation. We first decompose $\boldsymbol{A}(t)$ into an Hermitian part and an anti-Hermitian part:

$$\boldsymbol{A}(t) = \boldsymbol{H}_1(t) - i\boldsymbol{H}_2(t), \tag{2.2}$$

where

$$\boldsymbol{H}_1(t) = (\boldsymbol{A}(t) + \boldsymbol{A}^\dagger(t))/2 = \boldsymbol{H}_1^\dagger(t), \quad \boldsymbol{H}_2(t) = i(\boldsymbol{A}(t) - \boldsymbol{A}^\dagger(t))/2 = \boldsymbol{H}_2^\dagger(t)$$

are both Hermitian and we assume $\boldsymbol{H}_1$ to be negative semi-definite (so the original system is stable).

The key idea of Schrödingerization is to introduce a real one-dimensional variable $\xi > 0$ and define

$$\boldsymbol{w}(t,\xi) = e^{-\xi}\boldsymbol{u}(t). \tag{2.3}$$

This transformation is called the *warped phase transformation*. Clearly, $\boldsymbol{w}$ solves the following PDE:

$$\partial_t \boldsymbol{w} = -\boldsymbol{H}_1(t)\partial_\xi \boldsymbol{w} - i\boldsymbol{H}_2(t)\boldsymbol{w}. \tag{2.4}$$

Since $\boldsymbol{H}_1$ is negative semi-definite, the solution to (3.11) is a wave that moves *from right to left* in the $\xi$-domain, therefore one does not need a boundary condition at $\xi = 0$. The equation is stable and one can truncate the $\xi$-domain for a sufficiently large $\xi$ and impose a zero incoming boundary condition there to numerically well approximate the original problem.

One can recover the solution to the original equation for $\boldsymbol{u}(t)$ using

$$\boldsymbol{u}(t) = \int_0^\infty \boldsymbol{w}(t,\xi)\mathrm{d}\xi \quad \text{or} \quad \boldsymbol{u}(t) = e^{\xi}\boldsymbol{w}(t,\xi) \quad \text{for any} \quad \xi > 0. \tag{2.5}$$

To establish its connection to Schrödinger's type equation, we extend the domain of $\xi$ to $(-\infty,\infty)$, with evenly extended initial condition $\boldsymbol{w}(0,\xi) = \exp(-|\xi|)\boldsymbol{u}_0$. Let $\hat{\boldsymbol{w}} = \hat{\boldsymbol{w}}(t,\eta)$ be the Fourier transform of $\boldsymbol{w}$ in $\xi$ where $\eta \in \mathbb{R}$ is the Fourier mode:

$$\hat{\boldsymbol{w}}(t,\eta) = \int_{-\infty}^{\infty} e^{-i\xi\eta}\, \boldsymbol{w}(t,\xi)\, d\xi.$$

Then $\hat{\boldsymbol{w}}$ satisfies a system of *uncoupled* Schrödinger-like equations

$$i\partial_t \hat{\boldsymbol{w}} = (\eta \boldsymbol{H}_1(t) + \boldsymbol{H}_2(t))\hat{\boldsymbol{w}}, \tag{2.6}$$

one for each $\eta$! Clearly $\eta\boldsymbol{H}_1(t) + \boldsymbol{H}_2(t)$ *is* Hermitian, thus (2.6) is what we call the *Schrödingerised* equation for $\boldsymbol{u}$.

One can now design a quantum algorithm–digital or analog–starting from (2.6).

The initial value of $\boldsymbol{w}$ for $\xi < 0$ can be chosen to be in Sobolev space $H^k$, for any $k$, which will yield a smooth $\boldsymbol{w}$ so numerical approximation in the $\xi$-domain can be made to any order of accuracy [32], giving near optimal complexity (see [2], Section 2.3, about the issue of near optimality).

Next we give examples of parabolic PDEs that can be Schrödingerized. For more equations, such as the linear Boltzmann equation, and the Maxwell equation, see [31, 34].

For issues regarding to equations with time-dependent coefficients (non-autonomous system), see section 3.4.

## 2.2 The heat equation

For the heat equation

$$\partial_t u - \nabla_x^2 u = 0$$

the Schrödingeized equation is

$$i\partial_t \hat{w} = \eta \nabla_x^2 \hat{w}, \tag{2.7}$$

which is exactly the Schrödinger equation, for every Fourier mode $\eta$.

For elliptic equations, one can add the time-derivative to make it a parabolic equation and then evolve it to the steady state. See [21]. Likewise, such a time-marching strategy can be used to develop quantum algorithms for the preparation of quantum ground states or Gibbs states [37].

### 2.3 The Black-Scholes equation

The Black-Scholes equation

$$\partial_t V + rS\,\partial_S V + \frac{1}{2}\sigma^2 S^2\,\partial_{SS} V = rV,$$

is a PDE that evaluates the price of a financial derivative [17], where $r$ and $\sigma$ are constants. For a specific derivative contract, the problem is to determine its present price $V(t=0,S)$ according to the terminal price $V(t=T,S)$ of the option [17]. The change of variables $S = \mathrm{e}^x$, $-\infty < x < \infty$ and $t \to T - t$ leads to a forward parabolic equation

$$\partial_t V = (r - \frac{\sigma^2}{2})\partial_x V + \frac{\sigma^2}{2}\partial_{xx} V - rV. \tag{2.8}$$

By the warped phase transformation $W(t,x,p) = \mathrm{e}^{-p}V(t,x)$ with periodic extension of the initial data, one gets

$$\partial_t W = (r - \frac{\sigma^2}{2})\partial_x W + (\frac{\sigma^2}{2}\partial_{xx} - rI)(-\partial_p W).$$

Similar to the heat equation case, it is straightforward to derive a Hamiltonian system

$$i\partial_t \hat{W}(t) = \boldsymbol{H}\hat{W}(t),$$

$$\boldsymbol{H} = i\,(r - \frac{\sigma^2}{2})\partial_x + \eta\,(\frac{\sigma^2}{2}\partial_{xx} - rI).$$

Here $\boldsymbol{H}$ is a Hermitian operator.

### 2.4 The Fokker-Planck equation

The Fokker-Planck equation models the time evolution of the probability density function $f(t,x)$ of particles under the influence of drag and random forces [53]. It takes the form

$$\partial_t f = -\nabla \cdot (\nabla V f) + \sigma \Delta f, \tag{2.9}$$

where $V(x)$ is the scalar potential, and $\sigma > 0$ is the constant diffusion rate. The first term on the right-hand side is the drifted term, while the second term is the diffusion generated by the Brownian motion. The steady state solution of this equation is $f = \mathrm{e}^{-V(x)/\sigma}$. For convenience, we assume the periodic boundary conditions with $x = (x_1, \cdots, x_d) \in [-1,1]^d$.

#### 2.4.1 The conservation form

Equation (2.9) can also be written as

$$\partial_t f = \sigma \nabla \cdot \Big(\mathrm{e}^{-V/\sigma}\nabla\Big(\mathrm{e}^{V/\sigma} f\Big)\Big). \tag{2.10}$$

As done for the heat equation, one can introduce the transformation $F(t,x,p) = \mathrm{e}^{-p} f(t,x)$ and extend the initial data to $p < 0$ to obtain

$$\begin{cases} \partial_t F = \sigma \nabla_x \cdot \left( \mathrm{e}^{-V/\sigma} \nabla_x \left( \mathrm{e}^{V/\sigma} (-\partial_\xi F) \right) \right) \\ F(0,x,p) = \mathrm{e}^{-|p|} f(t,x). \end{cases}$$

Now a Fourier transform on $p$ gives the Schrödingerized form:

$$i \partial_t \hat{F} = \eta \sigma \, \nabla_x \cdot \left( \mathrm{e}^{-V/\sigma} \nabla_x \left( \mathrm{e}^{V/\sigma} \hat{F} \right) \right). \tag{2.11}$$

### 2.4.2 The heat equation form

Using the transformation $\psi(t,x) = \mathrm{e}^{V/(2\sigma)} f$, one obtains gets the following parabolic equation [50]

$$\partial_t \psi = \sigma \Delta \psi - U(x) \psi, \tag{2.12}$$

where

$$U(x) := \frac{|\nabla V(x)|^2}{4\sigma} - \frac{1}{2} \Delta V(x).$$

Since equation (2.12) has the same form of the heat equation, the Fourier transform of the warped phase transformation $\Psi = e^{-\xi} \psi$ gives

$$i \partial_t \hat{\Psi} = \eta [\sigma \, \Delta - U(x)] \hat{\Psi}. \tag{2.13}$$

# 3 Extensions

## 3.1 Inhomogeneous systems

Now we consider the inhomogeneous system:

$$\partial_t \boldsymbol{u} = A(t) \boldsymbol{u}(t) + \boldsymbol{b}(t), \quad \boldsymbol{u}(0) = \boldsymbol{u}_0, \tag{3.1}$$

where $\boldsymbol{u}$, $\boldsymbol{b} = (b_1, \cdots, b_n)^T \in \mathbb{C}^n$, $A \in \mathbb{C}^{n \times n}$ is a time-dependent matrix.

We first convert the inhomogeneous system to the homogeneous one by introducing $\tilde{A} \in \mathbb{C}^{2n \times 2n}$ with $\tilde{A} = \begin{bmatrix} A & B \\ O & O \end{bmatrix}$ with $B = \mathrm{diag}(b_1, \cdots, b_n)$ and $O$ is the $n \times n$ zero matrix. Then we expend $\boldsymbol{u}$ to $\boldsymbol{v} = \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{y} \end{bmatrix}$, to obtain

$$\partial_t \boldsymbol{v} = \tilde{A}(t) \boldsymbol{v}(t), \qquad \boldsymbol{v}(0) = \begin{bmatrix} \boldsymbol{u}_0 \\ \mathbf{1} \end{bmatrix}, \tag{3.2}$$

where $\mathbf{1}$ is the identity vector in which all components are 1.

The problem now is that when one does the decomposition (2.2), the matrix $\boldsymbol{H}_1$ will have *positive* eigenvlue [32]. This will induce *right moving wave* from the $\xi < 0$ domain to the $\xi > 0$ domain, which is non-physical and cannot be used when one recovers the original variable from the Schrödingerized system as in (2.5). The following theorem gives the way to deal with this issue:

**Theorem 3.1.** *[32] Assume the eigenvalues of $\boldsymbol{H}_1$ are $\lambda_j(\boldsymbol{H}_1)$, $(j = 1, \cdots, n)$, ordered monotonically increasing in $j$, with some of the eigenvalues positive. Then the solution of (3.1) can be recovered by*

$$\boldsymbol{u} = e^{\xi}\boldsymbol{w}(\xi), \quad \textit{for any } \xi \geq \xi^{\diamond}, \tag{3.3}$$

*where $\xi^{\diamond} \geq \max\{\lambda_n(\boldsymbol{H}_1)T, 0\}$, or recovered by using the integration,*

$$\boldsymbol{u} = e^{\xi^{\diamond}} \int_{\xi^{\diamond}}^{\infty} \boldsymbol{w}(\xi)\, d\xi. \tag{3.4}$$

The choice of $\xi^{\diamond}$ is to avoid using spurious data from the right-moving waves generated from domain $\xi < 0$.

## 3.2 Boundary value and interface problems

For PDEs with physical or artificial boundary conditions, one can discretize them spatially first, and then the boundary condition will appear in the inhomogeneous term $\boldsymbol{b}$ in (3.1). Then one can use the method described in section 3.1.

When modeling physical problems with heterogeneities, for example through different materials or media, one encounters interface problems where interface conditions need to be imposed. Examples include Stefan problem for parabolic equations [56] and wave propagations through different media where waves can be transmitted and reflected. Upon proper spatial discretizations of these PDEs and their interface conditions–in the spirit of immersed interface methods [39, 43], the interface conditions will also appear in the inhomogeneous term $\boldsymbol{b}$ in (3.1). Then one can again use the method described in section 3.1. See [23, 24].

## 3.3 Iterative solvers in numerical linear algebra

An iterative method in linear algebra (or a discrete linear dynamical system) can be written as

$$\boldsymbol{y}_{k+1} = G\boldsymbol{y}_k + \boldsymbol{g}, \qquad k \in \mathbb{Z}^+ \cup \{0\}, \tag{3.5}$$

where $\boldsymbol{y}_k, \boldsymbol{g} \in \mathbb{R}^n$ and $G$ is a $n \times n$ matrix. Here $k$ is the iteration step. The iteration converges if the spectral radius of $G$, $r(G) < 1$. (3.5) can be written as a homogeneous system by defining an augmented vector $\boldsymbol{x}_k = (\boldsymbol{y}_k, \mathbf{1})^T$:

$$\boldsymbol{x}_{k+1} = C\boldsymbol{x}_k, \qquad C = \begin{pmatrix} G & g \\ O & \mathbf{1} \end{pmatrix} \tag{3.6}$$

where $C$ is an $2n \times 2n$ matrix and $O$ is the $n \times n$ zero matrix. Rewrite this in the form

$$\boldsymbol{x}_{k+1} - \boldsymbol{x}_k = \begin{pmatrix} \boldsymbol{y}_{k+1} - \boldsymbol{y}_k \\ \mathbf{0} \end{pmatrix} = (C - I)\boldsymbol{x}_k. \tag{3.7}$$

One can convert $k$ into continuous time $t \in \mathbb{R}^+ \cup \{0\}$, so the iterative relation (3.7) is transformed into a corresponding system of linear ODEs

$$\frac{d\boldsymbol{x}}{dt} = (C - I)\boldsymbol{x}, \qquad \boldsymbol{x}(t=0) = \boldsymbol{x}_0. \tag{3.8}$$

Note that since $r(C) < 1$, then $C - I$ has *negative* eigenvalues. Hence the corresponding ODE system is contractive; thus the solution will decay to the steady state exponentially in $t$, which corresponds exactly to the convergence of the iterative method (3.5).

Now quantum simulation can be done by Schrödingerizing the ODE (3.8), see [30] for more details.

## 3.4 Non-autonomous systems

Consider a general linear non-autonomous dynamical system $\boldsymbol{u}(t)$ on a Hilbert space $\mathcal{H}$,

$$\frac{d\boldsymbol{u}(t)}{dt} = -i\boldsymbol{H}(t)\boldsymbol{u}(t) \qquad \boldsymbol{u}(0) = \boldsymbol{u}_0, \tag{3.9}$$

where $\boldsymbol{H}(t)$ is a linear (for ODEs) or linear differential (for PDEs) operator with time-dependent coefficients. Without loss of generality, we assume $\boldsymbol{H}$ is Hermitian: $\boldsymbol{H}(t) = \boldsymbol{H}^\dagger(t)$, which describes a quantum system evolving unitarily under a time-dependent Hamiltonian. Time-dependent Hamiltonians appear in many applications including adiabatic quantum computing [1, 3, 4]. For more general $\boldsymbol{H}$ one can use the Schrödingerisation to transform into the unitary dynamics.

The solution to Eq. (3.9) can be written as

$$\boldsymbol{u}(t) = \mathcal{U}_{t,0}\boldsymbol{u}_0, \tag{3.10}$$

where

$$\begin{aligned}\mathcal{U}_{t,s} = \mathcal{T}e^{-i\int_s^t \boldsymbol{H}(\tau)d\tau} &= \lim_{N\to\infty} e^{-i\boldsymbol{H}(t_N)\Delta t} \cdots e^{-i\boldsymbol{H}(t_1)\Delta t} \\ &= I + \sum_{n=1}^{\infty}(-i)^n \frac{1}{n!}\int_s^t dt_1 \cdots \int_s^t dt_n \mathcal{T}\boldsymbol{H}(t_1)\cdots \boldsymbol{H}(t_n),\end{aligned}$$

and $\mathcal{T}$ is the chronological time-ordering operator. The difficulty in developing a quantum algorithm here is that in general $\boldsymbol{H}(t)I$ is *non-cummutative*, namely $[\boldsymbol{H}(t), \boldsymbol{H}(t')] \neq 0$, thus one needs to evolve in time according to the time-dependent operator $\mathcal{T}\exp(-i\int_0^t \boldsymbol{H}(\tau)d\tau)$ chronologically [55]. For instance, in quantum simulation, not only are time-ordered oracles necessary, but also at each time-interval, a *different* Hamiltonian $\boldsymbol{H}(t_i)$, corresponding to different quantum gates, is required.

By introducing a new time variable $s$, the non-autonomous system can be converted into a new PDE system defined in one higher dimension, but with *time-independent* coefficients, so it becomes an autonomous system. This is realized by the following theorem [9, 19, 54, 57].

**Theorem 3.2.** *For the non-autonomous system in Eq.(3.9), consider the following initial-value problem of an autonomous PDE*

$$\begin{aligned}&\frac{\partial \boldsymbol{w}}{\partial t} + \frac{\partial \boldsymbol{w}}{\partial s} = -i\boldsymbol{H}(s)\boldsymbol{w} \\ &\boldsymbol{w}(0, s) = G(s)\boldsymbol{u}_0, \qquad s \in \mathbb{R}.\end{aligned} \tag{3.11}$$

*The analytical solution to this problem is*

$$\boldsymbol{w}(t,s) = G(s-t)\mathcal{U}_{s,s-t}\boldsymbol{y}_0, \qquad \mathcal{U}_{s,s-t} = \mathcal{T}e^{-i\int_{s-t}^{s}\boldsymbol{H}(\tau)d\tau} = \mathcal{T}e^{-i\int_0^t \boldsymbol{H}(s-t+\tau)d\tau}. \tag{3.12}$$

*When $G(s) = \delta(s)$, one can easily recover $\boldsymbol{u}(t)$ in Eq. (3.9) from $\boldsymbol{w}(t,s)$ using*

$$\boldsymbol{y}(t) = \int_{-\infty}^{\infty} \boldsymbol{w}(t,s)\, ds. \tag{3.13}$$

*Alternatively, when $G(s) = 1$, $\boldsymbol{u}(t)$ can be recovered with $\boldsymbol{y}(t) = \boldsymbol{w}(t, s=t)$.*

By standard quantization technique (see section 4.1), introducing the operators $s \to \hat{\boldsymbol{s}}$ and $\partial/\partial s \to i\hat{\boldsymbol{p}}_s$, then one can evolve the linear autonomous system Eq. (3.11) as

$$i\frac{d|\boldsymbol{w}\rangle}{dt} = \boldsymbol{H}|\boldsymbol{w}\rangle, \quad \boldsymbol{H} = \hat{\boldsymbol{p}}_s \otimes \boldsymbol{1} + \boldsymbol{H}(\hat{\boldsymbol{s}}) = \boldsymbol{H}^\dagger, \quad |\boldsymbol{w}\rangle(0) = \int ds\, G(s)|s\rangle \otimes |\boldsymbol{y}_0\rangle \tag{3.14}$$

where the Hamiltonian $\boldsymbol{H}$ is *time-independent*. This is called the Sambe-Howland's clock. As studied in [10], the Sambe-Howland clock can unify existing quantum algorithms for non-autonomous systems, either recovering the existing algorithms that correspond to different discretizations of the Sambe-Howland clock, or serving as the starting point to derive new and even more accurate algorithms.

## 3.5 Stochastic differential equations

Given a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, we first consider the following $d$-dimensional SDE with additive Gaussian noise:

$$dX_t = AX_t\, dt + B\, dW_t, \quad 0 \le t \le T; \quad X_0 = x_0. \tag{3.15}$$

Here, $W_t$ is the $d$-dimensional standard Brownian motion, $A$ and $B$ are $d \times d$ matrices.

We first discretize it by the Euler-Maruyama scheme. Let time step $t_k = kT/N_T$ with uniform time increment $\Delta t = T/N_T$, and $\{\widehat{X}(t_k)\}_{0\le k\le N}$ be the numerical solution to (3.15):

$$\widehat{X}(t_{k+1}) = \widehat{X}(t_k) + A\widehat{X}(t_k)\Delta t + B\Delta W_k, \quad k = 0,1,\dots,N_T-1; \quad \widehat{X}(t_0) = x_0, \tag{3.16}$$

where $\Delta W_k = W_{t_{k+1}} - W_{t_k}$ is a Gaussian random variable with mean 0 and covariance matrix $\Delta t I$.

Next, we replace the discrete time by a continuous time variable, to turn the evolution into an ODE equation for each time interval. Namely, for the temporal duration $(t_k, t_{k+1})$, we evolve the solution through the following *non-autonomous* ODE with inhomogeneous term:

$$\begin{cases} \frac{d}{dt}\widetilde{X}(t) = A\widetilde{X}(t) + \frac{B\Delta W_k}{\Delta t}, \quad t_k < t \le t_{k+1}, \quad k \ge 1, \\ \text{with } \widetilde{X}(t_k) \text{ calculated in the } (k-1)\text{-th iteration.} \end{cases} \tag{3.17}$$

Here $X(t_0) = x_0$, while $\Delta W_k = W_{t_{k+1}} - W_{t_k}$ is simulated by sampling independent $d$-dimensional normal distribution $\{\boldsymbol{\xi}_k : \boldsymbol{\xi}_k \sim N(0, \boldsymbol{I})\}$ :

$$\Delta W_k \overset{d}{=} \sqrt{\Delta t}\, \boldsymbol{\xi_k}.$$

The inhomogeneous term can be handled as described in Section 3.1.

Define

$$H_{1,k} = \begin{pmatrix} \frac{1}{2}(A + A^\dagger) & \frac{1}{2}B\Delta W_k/\sqrt{\Delta t} \\ \frac{1}{2}(\Delta W_k^\top/\sqrt{\Delta t})B^\dagger & 0 \end{pmatrix},$$
$$H_{2,k} = \begin{pmatrix} \frac{1}{2i}(A - A^\dagger) & -\frac{i}{2}B\Delta W_k/\sqrt{\Delta t} \\ \frac{i}{2}(\Delta W_k^\top/\sqrt{\Delta t})B^\dagger & 0 \end{pmatrix}.$$

Then the Schrödingerisation procedure can be applied for each $k$ to get the Schrödingerized system in the Fourier space (see [33]):

$$\begin{cases} i\frac{d}{dt}\hat{\boldsymbol{w}}(t,\eta) = \eta H_{1,k}\hat{\boldsymbol{w}}(t,\eta) + H_{2,k}\hat{\boldsymbol{w}}(t,\eta), \quad t_k < t \leq t_{k+1}, \\ \hat{\boldsymbol{w}}(0,\eta) = \frac{1}{\pi(1+\eta^2)}\widetilde{X}(0). \end{cases} \tag{3.18}$$

Equation (3.18) is a piecewise constant non-autonamous Schrödinger equation with Hamiltonian $\widetilde{H}(t)$ defined by

$$\widetilde{H}(t) = \sum_{k=0}^{N_T-1} \left(\eta H_{1,k} - H_{2,k}\right)\mathbb{1}_{(t_k,t_{k+1}]}(t).$$

The Schrödingerisation procedure can also be applied to linear SDE driven by $\alpha$-stable Lévy processes with $1 < \alpha < 2$. Consider a $d$-dimensional linear SDE driven by isotropic $\alpha$-stable Lévy processes:

$$dX_t = AX_t\,dt + BdL_t^\alpha, \quad X_0 = x_0, \quad 0 \leq t \leq T, \tag{3.19}$$

where $L_t^\alpha$ is a $d$-dimensional isotropic $\alpha$-stable Lévy process with $1 < \alpha < 2$ and it satisfies $L_t^\alpha \overset{d}{=} t^{\frac{1}{\alpha}}L_1$. Then we can proceed with the approximate equation

$$\begin{cases} \frac{d}{dt}\widetilde{X}(t) = A\widetilde{X}(t) + \frac{B\Delta L_k^\alpha}{\Delta t}, \quad t_k < t \leq t_{k+1}, \quad k \geq 1, \\ \text{with } \widetilde{X}(t_k) \text{ calculated in the } (k-1)\text{-th iteration.} \end{cases} \tag{3.20}$$

Here $X(t_0) = x_0$, and $\Delta L_k^\alpha = L_{t_{k+1}}^\alpha - L_{t_k}^\alpha \overset{d}{=} \Delta t^{\frac{1}{\alpha}}L_1$ are independent with identical stable distribution. Take random variables $\boldsymbol{\xi}_k \overset{d}{=} L_1$. The rest is similar to the Schrödingerization technique for the additive Gaussian noise case.

Similar treatment can also be applied to multiplicative Gaussian noise. Consider the following SDE:

$$dX(t) = AX(t)dt + \sum_{l=1}^m B^{(l)}X(t)dW_t^{(l)}, \quad X_0 = x_0. \tag{3.21}$$

Here, $W_t = (W_t^{(1)}, \cdots, W_t^{(m)})$ is a $m$-dimensional standard Brownian motion. $\{A, B^{(1)}, \cdots, B^{(m)}\}$ are $d \times d$ matrices.

For $t_k = kT/N_T$, we solve the following ODE:

$$\begin{cases} \dfrac{d\widetilde{X}(t)}{dt} = \widetilde{A}_k\widetilde{X}(t), \text{ and } t_k \leq t \leq t_{k+1}; \quad \widetilde{X}(0) = x_0, \\ \widetilde{A}_k = A - \dfrac{1}{2}\displaystyle\sum_{l=1}^m (B^{(l)})^2 + \sum_{l=1}^m B^{(l)}\frac{\Delta W_k^{(l)}}{\Delta t}, \end{cases} \tag{3.22}$$

where $\Delta W_k^{(l)} = W_{t_{k+1}}^{(l)} - W_{t_k}^{(l)}$ is a standard scalar Gaussian random variable. The term $-\frac{1}{2}\sum_{l=1}^m (B^{(l)})^2$ comes from the correction term from the Ito's formula. Then one can proceed with the Schrödingerization procedure as in the addition noise case. See [33].

# 4 Analog quantum computation

The difficulty in efficiently correcting errors due to noise that is present in currently available qubit-based quantum computers means that general-purpose digital quantum computers that can realise large-scale quantum algorithms with realistic applications are not possible in the very near term. The Schödingerization method, on the other hand, provides a framework that allows one to also explore *analog* quantum simulation as another platform to realise quantum algorithms. This is aimed toward developing quantum simulation devices that solve specific PDEs that *could* be realized in the near term. Compared to other quantum PDE solvers, which are usually based on temporal and/or spatial discretizations, Schrödingerization instead maps a continuous PDEs – via a continuous transformation – to another continuous PDEs. Thus it can be used to design continuous-variable, or *qumode*-based analog quantum simulators, running in continuous time, without the need of a large number of quantum gates. This can avoid or significantly alleviates the need for large-scale qubit-based devices for certain problems, as proposed in [27].

A continuous-variable (CV) quantum state, or *'qumode'*, is the quantum analogue of a continuous classical degree of freedom, like position, momentum or energy, before being quantised. It spans an infinite-dimensional Hilbert space, equipped with observables with a continuous spectrum, such as the position $\hat{x}$ and momentum $\hat{p} \leftrightarrow -i\partial/\partial x$, which are observables of a quantum particle. Choosing its eigenbasis as $\{|x\rangle\}_{x\in\mathbb{R}}$, which are the eigenstates of $\hat{x}$, then a qumode can be expressed as $|u\rangle = (1/\|\boldsymbol{u}\|)\int u(x)|x\rangle dx$, where $\|\boldsymbol{u}\|^2 = \int |u(x)|^2 dx$ is the normalisation constant. A system of $m$-qumodes is a tensor product of $m$ qumodes.

However, unlike other continuous-variable quantum algorithms (for example, see [8, 49]), our analog quantum algorithms keeps time continuous, in addition to continuous-variable quantum states. The precision also does not depend on the difficulty in creating suitable ancilla states (for example highly squeezed quantum states), unlike many other continuous-variable quantum algorithms based on the quantum circuit model [48] as well as the measurement-based quantum model [47].

## 4.1 Quantization of PDEs

For analog quantum simulation of PDEs, we quantize the Schrödingerized PDEs with $|x\rangle$ and $|p\rangle$, which are the eigenvectors of $\hat{x}$ and $\hat{p}$ respectively. This yields a quantum Hamiltonian to be used for physical experiments that correspond to such a Hamiltonian, which can be designed by employing more traditional techniques in quantum control.

Consider the $d$-dimensional general parabolic equation:

$$\frac{\partial u}{\partial t} - \sum_{i=1}^{d} \frac{\partial}{\partial x_i}\left(\sum_{j=1}^{d} D_{ij}(x_1, ..., x_d)\frac{\partial u}{\partial x_j}\right) = 0, \tag{4.1}$$

$$D_{ij}(x_1, ..., x_d) > 0, \tag{4.2}$$

where $D_{ij}$ form the (assumed to be symmtric ) diffusion matrix ($D_{ij} = D_{ji}$). This equation can be transformed into

$$\frac{d\boldsymbol{u}}{dt} = -\boldsymbol{A}\boldsymbol{u}, \qquad \boldsymbol{A} = \sum_{i,j=1}^{d} \hat{p}_i D_{ij}(\hat{x}_1, ..., \hat{x}_D)\hat{p}_j. \tag{4.3}$$

After the Schrödingerization, we get the following system

$$i\frac{d\hat{\boldsymbol{w}}}{dt} = -(\boldsymbol{A} \otimes \hat{\eta})\hat{\boldsymbol{w}}. \tag{4.4}$$

Suppose one has a constant diffusion coefficient $D_{ij}(x) = a\delta_{ij} \in \mathbb{R}$. The Hamiltonian $\boldsymbol{H}$ in this case generates an entangling gate, where $\boldsymbol{H} = -\sum_{j=1}^{d} a\hat{p}_j^2 \otimes \hat{\eta}$. Observe here that $\boldsymbol{H} = \sum_{j=1}^{d} \boldsymbol{H}_j$ where $[\boldsymbol{H}_i, \boldsymbol{H}_j] = 0, \forall\, i, j = 1, ..., d$, thus one can factorize $\exp(-i\boldsymbol{H}t) = \exp(-i\boldsymbol{H}_1 t)... \exp(-i\boldsymbol{H}_d t)$ where $\boldsymbol{H}_j = -a\hat{p}_j^2 \otimes \hat{\eta}$. Here the most challenging gate requires a pairwise third-order operator of the form $\exp(i\hat{p}^2 \otimes \hat{\eta}t)$, where $\hat{p}$ and $\hat{\eta}$ are quadrature operators. While in principle many quantum systems have these interaction terms, it remains challenging for existing devices.

Next we introduce an approximation of the parabolic equation by hyperbolic systems, which allows for easier experimental realization on various physical platforms, such as superconductors, trapped ions or Rydberg atoms, see [26].

## 4.2 The hyperbolic heat approximation

As an example, we approximate the solution to the $d$-dimensional heat equation

$$\frac{\partial \tilde{u}}{\partial t} = \sum_{j=1}^{d} k_j \frac{\partial^2 \tilde{u}}{\partial x_j}, \qquad \tilde{u}(0, x) = u(0, x), \qquad x = (x_1, \cdots, x_d), \tag{4.5}$$

using a $d$-dimensional hyperbolic heat equation. Here we allow $d$ distinct parameters $\epsilon_j \leq 11$, $j = 1, \cdots, d$, which can be chosen to take different values to allow greater flexibility for the the physical realisation. The system of $d+1$ hyperbolic heat equations (which is also called the telegraph equation, or Cattaneo or Goldstein-Taylor model) to be used to approximate (4.5) is [45]:

$$\frac{\partial u}{\partial t} = -\sum_{j=1}^{d} \frac{1}{\epsilon_j} \frac{\partial v_j}{\partial x_j} \tag{4.6}$$

$$\frac{\partial v_j}{\partial t} = -\frac{1}{\epsilon_j} \frac{\partial u}{\partial x_j} - \frac{1}{k_j \epsilon_j^2} v_j, \qquad j = 1, \cdots, d. \tag{4.7}$$

When $\epsilon_j \to 0$ for all $j$, the following approximation holds

$$v_j \to -k_j \epsilon_j \frac{\partial u}{\partial x_j}, \qquad j = 1, \cdots, d. \tag{4.8}$$

Inserting this into Eq. (4.6) allows one to recover the original heat equation (4.5).

Define the hybrid $d$-qumode and single *qudit* ($d$ level) quantum state

$$\boldsymbol{u}(t) = \int u(t, x)|x\rangle dx, \qquad \boldsymbol{v}_j(t) = \int v_j(t, x)|x\rangle dx,$$

$$|w(t)\rangle = \frac{\boldsymbol{w}(t)}{\|\boldsymbol{w}(t)\|}, \qquad \boldsymbol{w}(t) = |0\rangle \otimes \boldsymbol{u}(t) + \sum_{j=1}^{d} |j\rangle \otimes \boldsymbol{v}_j(t), \tag{4.9}$$

$$\|\boldsymbol{w}(t)\|^2 = \|\boldsymbol{u}(t)\|^2 + \sum_{j=1}^{d} \|\boldsymbol{v}_j(t)\|^2. \tag{4.10}$$

Although the $d$-level qudit can in principle be rewritten as a system of $\log d$ qubits, analog simulation on the qudit system is much more natural for our Hamiltonians since each interaction term is only between one qudit and one qumode. If instead one uses a system of qubits, then one would require multi-qubit entanglement, which is more difficult to experimentally realise.

Via $\partial/\partial x_j \to i\hat{p}_j$, one can rewrite Eqs. (4.6) and (4.7) as

$$\begin{aligned}\frac{d\boldsymbol{w}(t)}{dt} &= -i((|0\rangle\langle 1| + |1\rangle\langle 0|) \otimes \hat{p}_1/\epsilon_1 + (|0\rangle\langle 2| + |2\rangle\langle 0|) \otimes \hat{p}_2/\epsilon_2 \\ &\qquad + \cdots + (|0\rangle\langle d| + |d\rangle\langle 0|) \otimes \hat{p}_d/\epsilon_d)\boldsymbol{w}(t) \\ &\qquad - \left(\frac{1}{\epsilon_1^2 k_1}|1\rangle\langle 1| + \cdots \frac{1}{\epsilon_d^2 k_d}|d\rangle\langle d|\right) \otimes \mathbf{1}_x \boldsymbol{w}(t) = -i\boldsymbol{A}\boldsymbol{w}(t), \\ &\boldsymbol{A} = \boldsymbol{A}_1 - i\boldsymbol{A}_2 \\ &\boldsymbol{A}_1 = \sum_{j=1}^{d} \frac{1}{\epsilon_j}(|0\rangle\langle j| + |j\rangle\langle 0|) \otimes \hat{p}_j = \boldsymbol{A}_1^\dagger, \qquad \boldsymbol{A}_2 = \sum_{j=1}^{d} \frac{1}{\epsilon_j^2 k_j}|j\rangle\langle j| \otimes \mathbf{1}_x = \boldsymbol{A}_2^\dagger. \end{aligned} \tag{4.11}$$

Here an ancilla qumode is included with the initial state $|\Xi\rangle$, and we act on the total initial state $|w(0)\rangle \otimes |\Xi\rangle$.

For simplicity of experiment we can even choose the special initial conditions $v_j(0, x) = 0$. Note $t > \epsilon_j^2 \log(1/\epsilon_j)$ for all $j$, namely, beyond the initial layer, (4.8) is valid. Thus the initial condition we use to prepare $|w(0)\rangle = |0\rangle \otimes |u(0)\rangle$ becomes very simple.

Using Schrödingerisation, the Hamiltonian to simulate becomes

$$\boldsymbol{H} = \boldsymbol{A}_2 \otimes \hat{\eta} + \boldsymbol{A}_1 \otimes \mathbf{1}_\eta \tag{4.12}$$

$$= \sum_{j=1}^{d} \frac{1}{\epsilon_j^2 k_j}|j\rangle\langle j| \otimes \mathbf{1}_x \otimes \hat{\eta} + \sum_{j=1}^{d} \frac{1}{\epsilon_j}(|0\rangle\langle j| + |j\rangle\langle 0|) \otimes \hat{p}_j \otimes \mathbf{1}_\eta. \tag{4.13}$$

This initial state $|0\rangle \otimes |u(t)\rangle \otimes |\Xi\rangle$ consists of $d+1$ qumodes and a qudit with $d$ levels. For each interaction term, there is only a pairwise interaction between one qumode and one qudit at a time. Each interaction is again of the type $\sigma \otimes \hat{p}$, where $\sigma$ is a Pauli matrix. Specifically, the interaction terms are of the form of the electric dipole interaction $\sigma_x \otimes \hat{p}$ and magnetic dipole interaction $\sigma_z \otimes \hat{\eta}$, which can appear for instance in a Jaynes-Cummings-like model. Thus one just needs pairwise Jaynes-Cummings-like interactions between every two level system $|0\rangle, |j\rangle$ and a qumode, for every $j$, which is much easier to realize experimentally compared to working directly with the original heat equation with second order differential operators .

Other parabolic equations such as the Black-Scholes and Fokker-Planck equations can also be handled similarly. See [26] for more details.

## 5 Nonlinear PDEs and ODEs

Quantum computers are built using quantum mechanics principle, which is intrinsically *linear*, making it difficult to solve nonlinear problems. For nonlinear ODEs and PDEs, there are two ways to tackle this difficulty. One is to use *linear approximation*, by approximating the original nonlinear problem by linear system. A popular approach is the *Carleman* approximation [41, 46]. Since it approximates a nonlinear system by a finite dimensional linear system, its validity is restrictive, effective only for weak and special (such as quadratic) nonlinearity with strong dissipation. The other is *linear representation*, which seeks a higher dimensional representation which is linear, and the higher dimensional system is *equivalent*–thus without approximation–to the original problem. The representative methods for linear representation include the Koopman-von-Neumann method [13, 40] and the level set method [28, 35]. Comparing with the linear approximation methods, the linear representation methods introduce *no approximation* thus are valid for *all time* and any nonlinearity for the classes of ODEs and PDEs it solves. However this approach works for limited classes of PDEs which we will present below.

Next we summarize the level set methods for Hamilton-Jacobi equations and scalar hyperbolic balance laws. The latter includes nonlinear ODEs.

### 5.1 The Hamilton-Jacobi equation

Hamilton-Jacobi equations arise, for instance, in geometric optics, the semiclassical limit of the Schrödinger equation, the level set formulation of front propagation, etc. It has the following general form

$$\partial_t S^{[k]} + H(\nabla S^{[k]}, x) = 0, \quad t \in \mathbb{R}^+, x \in \mathbb{R}^d, S^{[k]}(t,x) \in \mathbb{R}, \tag{5.1}$$

$$S^{[k]}(0,x) = S_0^{[k]}(x), \quad k = 1, ..., M, \tag{5.2}$$

with $H$ being the Hamiltonian. Here we consider the problem of simulating the equation with $M$ different initial data, each indexed by $k$. We will show that the computational complexity of the proposed level set algorithm will be *independent of* $M$ if the average of the solution for $M$ different initial data is to be computed.

Two important classical examples of Hamiltonians are

$$H(x,p) = \frac{p^2}{2} + V(x), \tag{5.3}$$

corresponding to classical Newtonian particles with potential $V(x)$, and

$$H(x,p) = c(x)|p|, \tag{5.4}$$

that arises in geometric optics and the level set formulation of front propagation [15, 52], in which $c(x)$ is the reciprocal of the index of reflection or the speed of a propagating front in the normal direction.

Define $u^{[k]} = \nabla S^{[k]} \in \mathbb{R}^d$. Then $u^{[k]}$ solves a hyperbolic system of conservation laws in gradient form:

$$\partial_t u^{[k]} + \nabla H(u^{[k]}, x) = 0, \tag{5.5}$$

$$u^{[k]}(0,x) = \nabla S_0^{[k]}(x). \tag{5.6}$$

The level set function $\phi_i^{[k]}(t,x,p)$ can be defined by

$$\phi_i^{[k]}(t,x,p=u^{[k]}(t,x))=0, \tag{5.7}$$

where $i=1,\cdots,d$ and $x,p\in\mathbb{R}^d$, $k=1,...,M$. The *zero level set* of $\phi_i^{[k]}$ is the set $\{(t,x,p)|\phi_i^{[k]}(t,x,p)=0\}$.

Since $u^{[k]}(t,x)$ solves Eq. (5.5), then one can show that $\phi^{[k]}=(\phi_1^{[k]},\cdots,\phi_d^{[k]})\in\mathbb{R}^d$ solves a (linear!) Liouville equation [38]

$$\partial_t\phi^{[k]}+\nabla_p H\cdot\nabla_x\phi^{[k]}-\nabla_x H\cdot\nabla_p\phi^{[k]}=0. \tag{5.8}$$

Note that the (bi)-characteristics of the Liouville equation in Eq. (5.8) is the Hamiltonian system

$$\frac{\partial x}{\partial t}=\nabla_p H,\quad \frac{\partial p}{\partial t}=-\nabla_x H. \tag{5.9}$$

The initial data can be chosen as

$$\phi_i^{[k]}(0,x,p)=p_i-u_i^{[k]}(0,x),\quad i=1,\cdots,d. \tag{5.10}$$

Then $u^{[k]}$ can be recovered from the intersection of the zero level sets of $\phi_i$ $(i=1,\cdots,d)$, namely

$$u^{[k]}(t,x)=\{p(t,x)|\,\phi_i^{[k]}(t,x,p)=0,\, i=1,\cdots,d\}. \tag{5.11}$$

Note that $\phi_i^{[k]}(t,x,p)=0$ may have multiple (say $J_k$) roots, denoted by $p_\gamma(t,x)$ $(\gamma=1,\cdots,J_k)$, hence the so-called multi-valued solutions will arise [14, 22, 59], which are denoted by

$$u_\gamma^{[k]}(t,x)=\cap_{i=1}^d\{p_\gamma(t,x)|\phi_i^{[k]}(t,x,p_\gamma)=0\},\quad \gamma=1,\cdots,J_k. \tag{5.12}$$

Now a $(d+1)$-dimensional nonlinear PDE– the Hamilton-Jacobi equation– has been transformed to a $(2d+1)$-dimensional *linear* PDE –the Liouville equation–without *any* assumptions on either the form or extent of the original nonlinearity. Unlike the Carleman approximation approach [46], *no linear approximation is made*, nor the form of the nonlinearity ($H$ here) needs to be assumed. The mapping is *exact* and works for any $H$.

Finding the zero level set of $\phi_i^{[k]}$ $(i=1,\cdots d)$ is difficult for quantum algorithms. An alternative method is to solve for $\psi$, defined by the following problem

$$\partial_t\psi+\nabla_p H\cdot\nabla_x\psi-\nabla_x H\cdot\nabla_p\psi=0 \tag{5.13}$$

with the initial condition

$$\psi(0,x,p)=\frac{1}{M}\sum_{k=1}^M\prod_{i=1}^d\delta(p_i-u_i^{[k]}(0,x)). \tag{5.14}$$

Then it is easy to prove [25]

$$\psi(t,x,p)=\frac{1}{M}\sum_{k=1}^M\delta(\phi^{[k]}(t,x,p))\,. \tag{5.15}$$

An important observation here is that *all* $M$ distinct initial conditions of $u^{[k]}$ in the original problem have now been converted into a *single* initial condition in $\psi$. Thus cost for the corresponding quantum algorithm is *independent* of $M$.

It will be too costly to recover the classical point-wise value of $u$ for all mesh points from the quantum state of $\psi$. Often only the physical observables are of interest, which are defined by

$$\langle G(t,x)\rangle \equiv \int_{\mathbb{R}^d} G(p)\psi(t,x,p)dp = \frac{1}{M}\sum_{k=1}^{M}\int_{\mathbb{R}^d} G(p)\delta(\phi^{[k]}(t,x,p))dp. \tag{5.16}$$

This is an ensemble average of solution over the $M$ different initial data, which can be approximated by numerical quadrature rules [29].

For the Hamiltonian (5.3), the level set equation is the semi-classical limit of the Schrödinger equation, where the zeroth, first and second moments of $\psi$, correspond to the semi-classical classical limit to the position density (which is 1 in this case), current density and kinetic energy [25]. The total energy $E(t,x)$ can be recovered by combining the second and zeroth moments to choose $G(p) = (1/2)|p|^2 + V(x)$.

## 5.2 Scalar hyperbolic balance laws and nonlinear ODEs

Nonlinear hyperbolic balance laws arise for instance in gas dynamics, combustion, magnetohydrodynamics, shallow water and traffic flows. Here, we focus on *scalar* equation where $u^{[k]}(t,x) \in \mathbb{R}$ is a scalar solving an initial value problem of an $(d+1)$-dimensional first-order hyperbolic PDE with a nonlinear source term

$$\partial_t u^{[k]} + F(u^{[k]}) \cdot \nabla_x u^{[k]} + Q(x, u^{[k]}) = 0, \quad t \in \mathbb{R}^+, \quad x \in \mathbb{R}^d, \tag{5.17}$$

$$u^{[k]}(0,x) = u_0^{[k]}(x), \quad k = 1, ..., M. \tag{5.18}$$

Here the flux $F(u^{[k]}) : \mathbb{R} \to \mathbb{R}^d$ is a vector and $Q : \mathbb{R}^{d+1} \to \mathbb{R}$ is the nonlinear source term. This equation includes any such hyperbolic balance laws in conservative or non-conservative form, and nonlinear ODE (when $F = 0$).

Introduce the level set function $\phi(t,x,p)$ in $(d+1)+1 = d+2$ dimensions, where $p \in \mathbb{R}^1$. Its zero level set gives the solution $u$:

$$\phi^{[k]}(t,x,p) = 0 \quad \text{at} \quad p = u^{[k]}(t,x). \tag{5.19}$$

Then $\phi$ satisfies [38]

$$\partial_t \phi^{[k]} + F(p) \cdot \nabla_x \phi^{[k]} - Q(x,p)\partial_p \phi^{[k]} = 0, \tag{5.20}$$

$$\phi^{[k]}(0,x,p) = p - u_0^{[k]}(x). \tag{5.21}$$

Like for the Hamilton-Jacobi PDEs, we can similarly define a function $\psi$ by the following problem

$$\partial_t \psi + F(p) \cdot \nabla_x \psi - Q(x,p)\partial_p \psi = 0, \tag{5.22}$$

$$\psi(0,x,p) = \tfrac{1}{M}\textstyle\sum_{k=1}^{M} \delta(p - u_0^{[k]}(x)), \tag{5.23}$$

and one has

$$\psi(t,x,p)=\frac{1}{M}\sum_{k=1}^{M}\delta(\phi^{[k]}(t,x,p)). \tag{5.24}$$

The only difference compared to the Hamilton-Jacobi equation is that now $p\in\mathbb{R}^1$ instead of being a $d$-dimensional vector. This means the observable is now an integral over $\mathbb{R}$:

$$\langle g(t,x)\rangle=\int_{\mathbb{R}} g(p)\psi(t,x,p)dp. \tag{5.25}$$

It is important to remark that the level set equations for both Hamilton-Jacobi equations and scalar hyperbolic balance laws give multivalued solutions, not *viscosity solutions* [12, 42], beyond the formation of caustics or shocks. It remains open to develop efficient quantum algorithms–with quantum advantage–that can capture viscosity solutions to these semi-linear PDEs.

# 6 Final remarks and open problems

In this article, we briefly reviewed the Schrödingerization technique, introduced in [27, 34, 37], as a general methodology to develop quantum algorithms for linear partial and ordinary differential equations. It can also be employed for iterative methods in numerical linear algebra. This article is not meant to be comprehensive in reviewing all important results in quantum algorithms for PDEs and ODEs. Rather the focus is on the continuous-variable perspective which sets the foundation for analog quantum computation [27], and Schrodingerization is a means for allowing this type of quantum algorithm. This preserves the fully continuous nature of the ODEs and PDEs in both space and time. Other quantum algorithms for PDEs include, for example, unitarization of non-unitary dynamics by the Linear Combination of Hamiltonian simulation [5], methods using quantum linear algebra solvers [18] and using block encoding [6, 7, 11], which require discretization of the PDE. We also did not discuss quantum circuit designs using qubit-based algorithms [58], which can also be constructed starting from Schrödingerized equations, see for example [20, 36]. Although Schrodingerization can also be used on qubit-based quantum computers, we do not review these results here.

It is also important to point out that here we only focus on preparing quantum states of the final solution given an initial quantum state embedding the initial conditions of the ODE or PDE. We did not touch upon the issue of quantum measurements, which extracts classical data from the quantum state. We also did not discuss the preparation of the initial quantum states. These conversions between classical data and quantum states can be costly depending on the type of data and the information one wants to extract. We leave these discussions for future work and focus here only on the evolution process.

A generic mathematical approach used in our methodology is *dimension lifting*, also known as dilation. Through dimension lifting, we transformed nonlinear PDEs and ODEs to linear PDEs, non-autonomous systems to autonomous systems, and linear PDEs to Schrödinger-type PDEs. Each of these dimension-lifting technique is different. We can also transform PDEs and ODEs with random coefficients or inputs to deterministic PDEs with randomness appearing only in the initial data [16] and using a quantum version of stochastic galerkin methods [27]. While the curse-of-dimensionality is a major bottleneck for classical computers, it is not always so for quantum computers. Thus it is of significant mathematical interest to develop dimension-lifting techniques which can transform problems that cannot be simulated by quantum computers into ones that quantum computers can solve.

There remains many challenging open mathematical and algorithmic problems, especially for nonlinear problems. Here we list a few:

- Nonlinear ODEs. While the level set or Koopman-von Neumann method can transform them into linear PDEs, the dimension of linear PDEs equals the number of nonlinear ODEs, thus, the dimension is increased too much and there is no quantum advantage [28]. We need to develop new efficient quantum algorithms for systems of nonlinear ODEs with quantum advantage, which is not limited to either short evolution time or small nonlinearities.

- Scalar nonlinear PDEs. For scalar equations such as Hamilton-Jacobi or scalar hyperbolic balance laws – both are of transport type with first-order derivatives. The level set approach gives multivalued solutions. We need to develop efficient quantum algorithms for these equations that can capture the *viscosity solutions*. For other nonlinear PDEs that are not of transport type, it remains open to find linear representation methods to make them linear in higher – but not exponentially higher – dimension, or to find adequate approximations that work in strongly nonlinear regimes, with quantum advantage.

- Nonlinear systems of PDEs. The level set method does not directly apply here at the continuous level. In principle one can discretize first spatially so they become nonlinear ODEs and then the level set method can be applied. But like in the case of nonlinear ODEs, one won't obtain quantum advantage since the dimension of the resulting linear PDEs is too high [28].

## Acknowledgment

The authors acknowledge support by NSFC grant No. 12341104, the Shanghai Jiao Tong University 2030 Initiative, the Fundamental Research Funds for the Central Universities and the Science and Technology Commission of Shanghai Municipality (STCSM) grant no. 24LZ1401200 (21JC1402900). NL also acknowledges funding from the Science and Technology Program of Shanghai, China (21JC1402900) and NSFC grant No. 12471411.

The authors thank their collaborators whose joint works are discussed in this article, including Yu Cao, Francóis Golse, Junpeng Hu, Xiantao Li, Chuwen Ma, Wei Wei, Yue Yu, and Lei Zhang.

## References

[1] Tameem Albash and Daniel A. Lidar. Adiabatic quantum computation. *Rev. Mod. Phys.*, 90(1):015002, 2018.

[2] Dong An, Andrew M Childs, and Lin Lin. Quantum algorithm for linear non-unitary dynamics with near-optimal dependence on all parameters. *arXiv preprint arXiv:2312.03916*, 2023.

[3] Dong An, Di Fang, and Lin Lin. Time-dependent unbounded Hamiltonian simulation with vector norm scaling. *Quantum*, 5:459, 2021.

[4] Dong An and Lin Lin. Quantum Linear System Solver Based on Time-optimal Adiabatic Quantum Computing and Quantum Approximate Optimization Algorithm. *ACM Transactions on Quantum Computing*, 3(2):1–28, 2022.

[5] Dong An, Jin-Peng Liu, and Lin Lin. Linear combination of hamiltonian simulation for nonunitary dynamics with optimal state preparation cost. *Physical Review Letters*, 131(15):150603, 2023.

[6] D. W. Berry. High-order quantum algorithm for solving linear differential equations. *J. Phys. A: Math. Theor.*, 47(10):105301, 17 pp., 2014.

[7] D. W. Berry, A. M. Childs, A. Ostrander, and G. Wang. Quantum algorithm for linear differential equations with exponentially improved dependence on precision. *Comm. Math. Phys.*, 356(3):1057–1081, 2017.

[8] Samuel L Braunstein and Peter Van Loock. Quantum information with continuous variables. *Reviews of modern physics*, 77(2):513, 2005.

[9] Yu Cao, Shi Jin, and Nana Liu. Quantum simulation for time-dependent hamiltonians–with applications to non-autonomous ordinary and partial differential equations. *arXiv preprint arXiv:2312.02817*, 2023.

[10] Yu Cao, Shi Jin, and Nana Liu. A unifying framework for quantum simulation algorithms for time-dependent hamiltonian dynamics. *arXiv preprint arXiv:2411.03180*, 2024.

[11] Andrew M Childs, Jin-Peng Liu, and Aaron Ostrander. High-precision quantum algorithms for partial differential equations. *Quantum*, 5:574, 2021.

[12] Michael G Crandall and Pierre-Louis Lions. Viscosity solutions of hamilton-jacobi equations. *Transactions of the American mathematical society*, 277(1):1–42, 1983.

[13] Ilya Y Dodin and Edward A Startsev. On applications of quantum computing to plasma simulations. *Physics of Plasmas*, 28(9):092101, 2021.

[14] Björn Engquist and Olof Runborg. Multi-phase computations in geometrical optics. *Journal of Computational and Applied Mathematics*, 74(1-2):175–192, 1996.

[15] Björn Engquist and Olof Runborg. Computational high frequency wave propagation. *Acta numerica*, 12:181–266, 2003.

[16] Franćois Golse, Shi Jin, and Nana Liu. Quantum algorithms for uncertainty quantification: application to partial differential equations. *arXiv preprint arXiv:2209.11220*, 2022.

[17] Javier Gonzalez-Conde, Angel Rodriguez-Rozas, Enrique Solano, and Mikel Sanz. Simulating option price dynamics with exponential quantum speedup. *arXiv:2101.04023v3*, 2022.

[18] A. W. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103(15):150502, 4 pp., 2009.

[19] James S. Howland. Stationary scattering theory for time-dependent Hamiltonians. *Mathematische Annalen*, 207(4):315–335, December 1974.

[20] Junpeng Hu, Shi Jin, Nana Liu, and Lei Zhang. Quantum circuits for partial differential equations via schrödingerisation. *arXiv preprint arXiv:2403.10032*, 2024.

[21] Junpeng Hu, Shi Jin, and Lei Zhang. Quantum algorithms for multiscale partial differential equations. *Multiscale Model. Simul.*, 22(3):1030–1067, 2024.

[22] Shi Jin and Xiantao Li. Multi-phase computations of the semiclassical limit of the Schrödinger equation and related problems: Whitham vs. Wigner. *Phys. D*, 182(1-2):46–85, 2003.

[23] Shi Jin, Xiantao Li, Nana Liu, and Yue Yu. Quantum simulation for partial differential equations with physical boundary or interface conditions. *Journal of Computational Physics*, 498:112707, 2024.

[24] Shi Jin, Xiantao Li, Nana Liu, and Yue Yu. Quantum simulation for quantum dynamics with artificial boundary conditions. *SIAM Journal on Scientific Computing*, 46(4):B403–B421, 2024.

[25] Shi Jin, Hailiang Liu, Stanley Osher, and Yen-Hsi Richard Tsai. Computing multivalued physical observables for the semiclassical limit of the Schrödinger equation. *J. Comput. Phys.*, 205(1):222–241, 2005.

[26] Shi Jin and Nana Liu. Analog quantum simulation of parabolic partial differential equations using jaynes-cummings-like models. *arXiv preprint arXiv:2407.01913*, 2024.

[27] Shi Jin and Nana Liu. Analog quantum simulation of partial differential equations. *Quantum Science and Technology*, 9(3):035047, 2024.

[28] Shi Jin and Nana Liu. Quantum algorithms for nonlinear partial differential equations. *Bulletin des Sciences Mathématiques*, 194:103457, 2024.

[29] Shi Jin and Nana Liu. Quantum algorithms for nonlinear partial differential equations. *Bull. Sci. Math.*, 194:Paper No. 103457, 66, 2024.

[30] Shi Jin and Nana Liu. Quantum simulation of discrete linear dynamical systems and simple iterative methods in linear algebra. *Proceedings of the Royal Society A*, 480(2291):20230370, 2024.

[31] Shi Jin, Nana Liu, and Chuwen Ma. Quantum simulation of maxwell's equations via schrödingerisation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 2023.

[32] Shi Jin, Nana Liu, and Chuwen Ma. On schrödingerization based quantum algorithms for linear dynamical systems with inhomogeneous terms. *preprint*, 2024.

[33] Shi Jin, Nana Liu, and Wei Wei. Quantum algorithms for stochastic differential equations: A schröingerisation approach. *arXiv preprint arXiv:2412.14868*, 2024.

[34] Shi Jin, Nana Liu, and Yue Yu. Quantum simulation of partial differential equations: Applications and detailed analysis. *Physical Review A*, 108(3):032603, 2023.

[35] Shi Jin, Nana Liu, and Yue Yu. Time complexity analysis of quantum algorithms via linear representations for nonlinear ordinary and partial differential equations. *Journal of Computational Physics*, 487:112149, 2023.

[36] Shi Jin, Nana Liu, and Yue Yu. Quantum circuits for the heat equation with physical boundary conditions via schrodingerisation. *arXiv preprint arXiv:2407.15895*, 2024.

[37] Shi Jin, Nana Liu, and Yue Yu. Quantum simulation of partial differential equations via schrödingerization. *Physical Review Letters*, 133(23):230602, 2024.

[38] Shi Jin and Stanley Osher. A level set method for the computation of multivalued solutions to quasi-linear hyperbolic PDEs and Hamilton-Jacobi equations. *Commun. Math. Sci.*, 1(3):575–591, 2003.

[39] Shi Jin and Xin Wen. A hamiltonian-preserving scheme for the liouville equation of geometrical optics with partial transmissions and reflections. *SIAM journal on numerical analysis*, 44(5):1801–1828, 2006.

[40] Ilon Joseph. Koopman–von neumann approach to quantum simulation of nonlinear classical dynamics. *Physical Review Research*, 2(4):043102, 2020.

[41] Hari Krovi. Improved quantum algorithms for linear and nonlinear differential equations. *Quantum*, 7:913, 2023.

[42] Peter D Lax. *Hyperbolic systems of conservation laws and the mathematical theory of shock waves*. SIAM, 1973.

[43] Randall J LeVeque and Zhilin Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis*, 31(4):1019–1044, 1994.

[44] Lin Lin. Lecture notes on quantum algorithms for scientific computation. *arXiv preprint arXiv:2201.08309*, 2022.

[45] Pierre-Louis Lions and Giuseppe Toscani. Diffusive limit for finite velocity boltzmann kinetic models. *Revista Matematica Iberoamericana*, 13(3):473–513, 1997.

[46] Jin-Peng Liu, Herman Oie Kolden, Hari K Krovi, Nuno F Loureiro, Konstantina Trivisa, and Andrew M Childs. Efficient quantum algorithm for dissipative nonlinear differential equations. *Proceedings of the National Academy of Sciences*, 118(35):e2026805118, 2021.

[47] Nana Liu, Tommaso F Demarie, Si-Hui Tan, Leandro Aolita, and Joseph F Fitzsimons. Client-friendly continuous-variable blind and verifiable quantum computing. *Physical Review A*, 100(6):062309, 2019.

[48] Nana Liu, Jayne Thompson, Christian Weedbrook, Seth Lloyd, Vlatko Vedral, Mile Gu, and Kavan Modi. Power of one qumode for quantum computation. *Physical Review A*, 93(5):052304, 2016.

[49] Seth Lloyd and Samuel L Braunstein. Quantum computation over continuous variables. *Physical Review Letters*, 82(8):1784, 1999.

[50] P. A. Markowich and C. Villani. On the trend to equilibrium for the Fokker-Planck equation: an interplay between physics and functional analysis. volume 19, pages 1–29. 2000. VI Workshop on Partial Differential Equations, Part II (Rio de Janeiro, 1999).

[51] Michael A. Nielsen and Isaac L. Chuang. *Quantum computation and quantum information.* Cambridge University Press, Cambridge, 2000.

[52] Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.

[53] Grigorios A Pavliotis. The fokker–planck equation. *Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations*, pages 87–137, 2014.

[54] Uri Peskin and Nimrod Moiseyev. The solution of the time-dependent Schrödinger equation by the $(t, t')$ method: Theory, computational algorithm and applications. *The Journal of Chemical Physics*, 99(6):4590–4596, September 1993.

[55] Michael Reed and Barry Simon. *Methods of mathematical physics. Fourier analysis, self-adjointness.* Academic Press, 1975.

[56] LI Rubinshtein. *The Stefan Problem*, volume 27. American Mathematical Soc., 1971.

[57] Hideo Sambe. Steady States and Quasienergies of a Quantum-Mechanical System in an Oscillating Field. *Physical Review A*, 7(6):2203–2213, 1973.

[58] Yuki Sato, Ruho Kondo, Ikko Hamamura, Tamiya Onodera, and Naoki Yamamoto. Hamiltonian simulation for hyperbolic partial differential equations by scalable quantum circuits. *Physical Review Research*, 6(3):033246, 2024.

[59] C. Sparber, P. Markowich, and N. Mauser. Wigner functions versus WKB-methods in multivalued geometrical optics. *Asymptot. Anal.*, 33(2):153–187, 2003.