J. Plasma Phys. (2025), *vol*. 91, E55 © The Author(s), 2025. Published by Cambridge University Press.
This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited. doi:10.1017/S0022377825000091

Quantum simulation of nonlinear dynamical systems using repeated measurement

Joseph Andress[1], Alexander Engel[1], Yuan Shi[1] and Scott Parker[1,2]

[1]Department of Physics, Center for Integrated Plasma Studies, University of Colorado Boulder, Boulder, CO 80309, USA

[2]Renewable and Sustainable Energy Institute, University of Colorado Boulder, Boulder, CO 80309, USA

Corresponding author: Joseph Andress, joan1465@colorado.edu

(Received 23 September 2024; revision received 21 January 2025; accepted 22 January 2025)

We present a quantum algorithm based on repeated measurement to solve initial-value problems for nonlinear ordinary differential equations (ODEs), which may be generated from partial differential equations in plasma physics. We map a dynamical system to a Hamiltonian form, where the Hamiltonian matrix is a function of dynamical variables. To advance in time, we measure expectation values from the previous time step and evaluate the Hamiltonian function classically, which introduces stochasticity into the dynamics. We then perform standard quantum Hamiltonian simulation over a short time, using the evaluated constant Hamiltonian matrix. This approach requires evolving an ensemble of quantum states, which are consumed each step to measure the required observables. We apply this approach to the classic logistic and Lorenz systems, in both integrable and chaotic regimes. Our analysis shows that the solutions' accuracy is influenced by both the stochastic sampling rate and the nature of the dynamical system.

Key words: quantum plasma, plasma nonlinear phenomena, plasma simulation

1. Introduction

As modern quantum computers move closer to true utility, increasing focus has come onto the various tasks a hypothetical error-corrected quantum computer might be able to perform. The existence of an efficient quantum algorithm for solving general dynamical systems of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{G}(\boldsymbol{x}) \tag{1.1}$$

is an open question, which has attracted significant attention (Leyton & Osborne 2008; Joseph 2020; Lubasch *et al.* 2020; Lloyd *et al.* 2020; Engel, Smith & Parker 2021; Liu *et al.* 2021; Xue, Wu & Guo 2021; Shi *et al.* 2021; Joseph *et al.* 2023; Shi *et al.*, 2024*b*). While previous approaches, which include variational methods (Lubasch *et al.* 2020), Carleman linearisation (Liu *et al.* 2021) and homotopy

perturbation methods (Xue *et al.* 2021), have shown promise in specific cases, none has yet achieved general efficiency, in the sense of a large quantum speedup with respect to the system size for general or polynomial dynamical systems without a complexity that is exponential in simulation time. For instance, the approach proposed by Leyton & Osborne (2008) achieves a speedup with respect to system size for a large class of polynomial dynamical systems through a non-deterministic method based on Euler's method, but has complexity exponential in simulation time. The algorithm of Liu *et al.* (2021) can scale polynomially with the simulation time, but requires that the dynamical system has dissipation with strength above a certain threshold, which unfortunately excludes many systems of interest such as collisionless plasma systems.

In the special case that $\boldsymbol{G}(\boldsymbol{x})$ is linear and norm-preserving, (1.1) may be mapped to Schrödinger's equation

$$\frac{d}{\mathrm{d}t}|\psi\rangle = -i\boldsymbol{H}|\psi\rangle \tag{1.2}$$

and thus solutions may be extracted from established Hamiltonian simulation algorithms (Childs & Wiebe 2012). These algorithms are themselves not always efficient, but extensive research into Hamiltonian simulation has uncovered efficient algorithms for a number of classes of $\boldsymbol{H}$ (Berry *et al.* 2014, 2015). In fact, many linearised plasma problems can be converted into the Schrödinger form (Engel, Smith & Parker 2019; Dodin & Startsev 2021).

However, what makes many plasma problems challenging are nonlinearities. For example, in the Vlasov–Maxwell system, nonlinearity arises from the coupling between the particle distribution function and the electromagnetic fields. The nonlinearity of this equation, central to plasma physics, is a major hurdle in applying quantum computing to the field. The Vlasov–Maxwell system is a set of partial differential equations, but methods for ordinary differential equations would be applicable after discretising the equation. For example, using finite difference in phase space (McLachlan 2003), the solution is specified only at a set of grid points, and derivatives are approximated as functions of the solution. After discretisation, the remaining task is to perform time advance.

In this paper, we present a quantum algorithm for advancing nonlinear ordinary differential equations (ODEs), whose only requirements are that the dynamical system is real and the nonlinearities are polynomial. Our algorithm uses a measurement-based approach on an ensemble of quantum states. We evolve each realisation of the ensemble separately and measure it to determine a snapshot of the nonlinear Hamiltonian. We then use Hamiltonian simulation to advance the system forward a short period, before again measuring the system to determine a new Hamiltonian. We demonstrate our algorithm by solving the logistic equation and the widely studied Lorentz system (Lorenz 1963). Even though the algorithm presented here is relatively simple, it can solve nonlinear problems for which previous quantum algorithms fail (Sanavio *et al.* 2024). Our goal is to demonstrate how an easy-to-understand quantum algorithm can solve a real problem, building a foundation for problems of practical interest in plasma science, such as particle dynamics in non-uniform magnetic fields (Kabin 2021), in the future. Although not generally efficient, our measurement-based approach may efficiently solve classes of problems not covered by existing quantum algorithms.

This paper is organised as follows. In § 2, we outline the basic algorithm and present a mapping which takes a real, polynomial system to Hamiltonian form.

https://doi.org/10.1017/S0022377825000091 Published online by Cambridge University Press

In § 3, we demonstrate the algorithm using two dynamical systems, the logistic system and the Lorenz system, and we examine the algorithm's performance under the lens of quantum entropy and error. Finally, in § 4, we summarize our findings. Additional details on the mapping in § 2 can be found in Appendix A.

2. Approach

In this paper, we attempt to generalise established methods of Hamiltonian simulation to real, polynomial systems of ODEs. Existing Hamiltonian simulation methods evolve a quantum state under a constant Hamiltonian, but by dividing the total simulation into many short times steps, we can use a different Hamiltonian for each step. Thus, the requirement of a constant Hamiltonian can be relaxed. In this paper, we generalise a constant Hamiltonian to a sum of constant Hamiltonians weighted by dynamical observables. Because the observables depend quadratically on quantum states, the Schrödinger equation has a cubic nonlinearity. A system of ODEs can be mapped directly to cubic, nonlinear Hamiltonian form if the ODEs are cubic and norm-preserving. More generally, we present a method of mapping any real, polynomial system to cubic, norm-preserving form.

The non-deterministic nature of quantum measurements causes uncertainties in the measured quadratic observables, resulting in a stochastic spreading of the possible simulated trajectories. Using many measurements to reduce the variance, the ensemble of many such trajectories converges to the deterministic solution, and we use von Neumann entropy as a measure of this spreading.

2.1. *Piecewise linear dynamics via measurements*

For dynamical systems of the form (1.1), we encode elements of the vector $\boldsymbol{x}$ using amplitudes of a multi-level quantum system ψ:

$$|x_i|^2 \propto |\langle i|\psi\rangle|^2. \tag{2.1}$$

This amplitude encoding scheme allows 2^N real scalar dynamical variables to be stored on N qubits. We use an approach akin to the classical forward Euler method. For the nth time step, we evolve the quantum state forward by Δt via (1.2) using a constant Hamiltonian:

$$|\psi\rangle_n = \exp\left(-i\boldsymbol{H}(|\psi\rangle_{n-1})\Delta t\right)|\psi\rangle_{n-1}. \tag{2.2}$$

In this unitary map, we allow $\boldsymbol{H}$ to be a function of $|\psi\rangle_{n-1}$, which we evaluate using measurements before evolving the state. Since the Hamiltonian is a constant over the time step, we can make use of linear Hamiltonian simulation algorithms. These algorithms are most efficient for sparse Hamiltonians (Berry *et al.* 2014, 2015). For more general Hamiltonians, one may use Trotter–Suzuki methods (Suzuki 1991) to partition the Hamiltonian into the sum of multiple sub-Hamiltonians.

2.2. *Observable Hamiltonian pairs for cubic nonlinear systems*

For (2.2) to be applicable for a nonlinear dynamical system, the system must be expressible in a Hamiltonian form. The simplest nonlinear example is perhaps the cubic system, for which $\boldsymbol{H}(|\psi\rangle_{n-1})$ can be written as the sum of multiple sub-Hamiltonians, weighted by the expectation values of the corresponding observables:

https://doi.org/10.1017/S0022377825000091 Published online by Cambridge University Press

$$\boldsymbol{H}(|\psi\rangle) = \sum_{k=1}^{M} \langle\psi|\boldsymbol{O}_k|\psi\rangle \boldsymbol{H}_k. \tag{2.3}$$

Because the observables are quadratic in $|\psi\rangle$, the dynamics $i\partial_t|\psi\rangle = \boldsymbol{H}(|\psi\rangle)|\psi\rangle$ is cubic in $|\psi\rangle$. For a given system, the observable-Hamiltonian pairs $\{\boldsymbol{O}_k, \boldsymbol{H}_k\}$ are constant. In other words, the pair does not depend on the quantum state, which means that they need be calculated only once before actual evolution begins. As an example, (3.6) gives the observable Hamiltonian pair associated for the logistic system.

Because measurement collapses the state, evaluating $\boldsymbol{H}$ before each time step requires many separate copies of $|\psi\rangle_{n-1}$. By the no-cloning theorem, a specific quantum state cannot be copied, but if the initial state and consequent evolution are stored classically, then the state can be recreated. In other words, it is impossible to directly copy $|\psi\rangle_{n-1}$ before evaluating $\boldsymbol{H}$; each copy of $|\psi\rangle_{n-1}$ must be evolved from $|\psi\rangle_0$.

Simulating to final time T with time steps of Δt requires $T/\Delta t$ separate time steps, and with m measurements for each of the M observables, a total of $mMT/\Delta t$ states will be consumed to evaluate $\boldsymbol{H}$ throughout the full simulation. However, it is possible to avoid storing more than one quantum state at a time by evolving each of the states separately and using classical memory to store the sampled expectation values required to recreate $\boldsymbol{H}$. This approach would require classical communication channels and classical memory to store and transfer the results of quantum measurement (Yepz 2001). This scheme is efficient if the dimension of $\boldsymbol{H}_k$ is large, but the number of observables M is small.

2.3. *Generalising to polynomial nonlinearities: enforcing homogeneity*

The requirement that the dynamical system be of the cubic Hamiltonian form can be relaxed, because any real system with polynomial nonlinearity can be mapped to the cubic Hamiltonian form. The first step of the mapping is to express the polynomial system in the tensor form

$$\dot{\boldsymbol{x}} \equiv \boldsymbol{A}\boldsymbol{x}^{\otimes q}. \tag{2.4}$$

In index notation, the above equation can be written more explicitly as

$$\dot{x}_j = \sum_{\alpha} \delta_{j\alpha_1} \boldsymbol{A}_\alpha \prod_{k=2}^{q+1} x_{\alpha_k}, \tag{2.5}$$

where each α is a multi-index containing $q+1$ indices. In (2.5), the Kronecker delta ensures that the only terms contributing to the derivative of x_j have the index j as the first entry of α, while the remaining terms in (2.5) account for the coefficient $\boldsymbol{A}_\alpha$ and factors in $\boldsymbol{x}$ for this specific monomial term in the degree-q polynomial. By this definition, all entries of $\boldsymbol{A}$ are labelled by $q+1$ indices and any entry with first index α_1 contributes a monomial term to the polynomial derivative of x_{α_1}. The coefficients for these monomials are stored in $\boldsymbol{A}$ and the remaining q indices correspond to the q factors of $\boldsymbol{x}$ entries in the degree-q monomial.

The form of (2.4) describes polynomial systems with homogeneous degree q. For non-homogeneous systems, we can homogenise (1.1) by adding a constant

https://doi.org/10.1017/S0022377825000091 Published online by Cambridge University Press

coordinate x_0 as the first component of $\boldsymbol{x}$:

$$x_0 = c. \tag{2.6}$$

We multiply this constant term into lower-degree terms to raise them to degree q. To keep the constant term unchanged during time evolution, we add a trivial term to the dynamical system:

$$\dot{x}_0 = G_0(\boldsymbol{x}) = 0. \tag{2.7}$$

For example, in (3.2), the logistic system is raised to an homogeneous degree 3. The exact value chosen for $x_0 = c$ is arbitrary, but it should be chosen to match the general scale of the other components of $\boldsymbol{x}$. In other words, it should be comparable to the other amplitudes of the normalised quantum state.

The tensor form of a system is generally not unique, as a result of the commutativity of multiplication. For example, consider the differential equation $\dot{x}_1 = 5x_1x_2x_3$. When encoding this equation into the tensor form $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}^{\otimes 3}$, the most general statement that can be made about $\boldsymbol{A}$ is then $\boldsymbol{A}_{1123} + \boldsymbol{A}_{1132} + \boldsymbol{A}_{1213} + \boldsymbol{A}_{1231} + \boldsymbol{A}_{1312} + \boldsymbol{A}_{1321} = 5$, because the order of factors x_{α_k} in (2.5) has no effect on the underlying dynamics.

2.4. *Rescaling to unitary evolution: enforcement of norm-preservation*

The second step of the mapping is to rescale the dynamical system. In general, a solution to (1.1) cannot be encoded into a quantum state per (2.1) without losing information on the magnitude of $\boldsymbol{x}$, because storage within a quantum state requires normalisation. It is therefore necessary to find a solution to the normalised problem,

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{F}(\hat{\boldsymbol{x}}), \tag{2.8}$$

where $\hat{\boldsymbol{x}} = \boldsymbol{x}/|\boldsymbol{x}|$, while also storing the overall scale of the solution. The dynamics of such a solution would be compatible with the normalisation of a quantum state while allowing the exact classical trajectory to be reconstructed.

An homogeneous degree-q polynomial in $\boldsymbol{x}$ scales with a^q when evaluated on $a\boldsymbol{x}$. Suppose the dynamical system (1.1) has already been converted to the homogeneous tensor form (2.4), then

$$\boldsymbol{G}(\boldsymbol{x}) = |\boldsymbol{x}|^q \boldsymbol{G}(\hat{\boldsymbol{x}}). \tag{2.9}$$

Taking the time derivative of $\hat{\boldsymbol{x}} = \boldsymbol{x}/|\boldsymbol{x}|$, the dynamics of the normalised solution satisfy

$$\dot{\hat{\boldsymbol{x}}} = |\boldsymbol{x}|^{q-1}\left(|\hat{\boldsymbol{x}}|^2 \boldsymbol{G}(\hat{\boldsymbol{x}}) - [\hat{\boldsymbol{x}} \cdot \boldsymbol{G}(\hat{\boldsymbol{x}})]\hat{\boldsymbol{x}}\right) = |\boldsymbol{x}|^{q-1}\boldsymbol{F}(\hat{\boldsymbol{x}}). \tag{2.10}$$

In the above equation, although $|\hat{\boldsymbol{x}}|^2 = 1$, we keep this term so that $\boldsymbol{F}(\hat{\boldsymbol{x}})$ is an homogeneous polynomial of degree $q+2$. The term in the bracket is the projection operator $\delta^{ij} - \hat{x}^i\hat{x}^j$ acting on $\boldsymbol{G}$, which removes the dynamics parallel to $\boldsymbol{x}$ so that the norm of the vector is preserved. The above equation converts the original system $\dot{\boldsymbol{x}} = \boldsymbol{G}(\boldsymbol{x})$ to a norm-preserving system $\dot{\hat{\boldsymbol{x}}} = \boldsymbol{F}(\hat{\boldsymbol{x}})$. A derivation of (2.10) can be found in Appendix A. Allowing for a nonlinear mapping in time, we can write

$$\begin{aligned} \frac{d}{\mathrm{d}t'}\hat{\boldsymbol{x}} &= \boldsymbol{F}(\hat{\boldsymbol{x}}), \\ \frac{\mathrm{d}t}{dt'} &= |\boldsymbol{x}|^{1-q}, \end{aligned} \tag{2.11}$$

https://doi.org/10.1017/S0022377825000091 Published online by Cambridge University Press

to hide the prefactor in (2.10). In other words, the dynamics that is parallel to $\boldsymbol{x}$ is captured as a rescaling in time. The scaling prefactor is well defined as long as $|\boldsymbol{x}|$ is non-zero. This is guaranteed when we add the constant variable x_0 in (2.6), which allows for the recovery of $|\boldsymbol{x}|$ from the normalised solution, and the constant serves as a non-zero minimum on $|\boldsymbol{x}|$. In the case of the simple linear system $\dot{x}_1 = \lambda x_1$, following the addition of constant x_0, (2.10) yields $\dot{\hat{x}}_0 = -\lambda \hat{x}_0 \hat{x}_1^2$, $\dot{\hat{x}}_1 = +\lambda \hat{x}_0^2 \hat{x}_1$ with $\partial_t \hat{\boldsymbol{x}} = \partial_{t'} \hat{\boldsymbol{x}}$, because $q = 1$. In other words, although for the original linear system, x_1 depends exponentially on time, our general procedure allows it to be embedded into a norm-preserving system.

We can write the homogeneous, degree-$(q+2)$ system in (2.11) using a rank-$(q+3)$ tensor:

$$\frac{d}{\mathrm{d}t'} \hat{\boldsymbol{x}} \equiv \boldsymbol{F} \hat{\boldsymbol{x}}^{\otimes (q+2)}. \tag{2.12}$$

In other words, we introduce a tensor $\boldsymbol{F}$ corresponding to the homogeneous polynomial $\boldsymbol{F}$ such that $\boldsymbol{F}(\hat{\boldsymbol{x}}) = \boldsymbol{F} \hat{\boldsymbol{x}}^{\otimes (q+2)}$. The tensor form is more easily analysed, manipulated and mapped to matrices, such as an observable or Hamiltonian, than the polynomial form. As an example, (3.4) gives the tensor form of the logistic system.

2.5. *Converting to Hamiltonian dynamics: enforcement of symmetry*

Mapping the tensor $\boldsymbol{F}$ to Hamiltonian form requires that $\boldsymbol{F}$ be antisymmetric in its first two indices. The entries of $\boldsymbol{F}$ must correspond to the entires of $\boldsymbol{H}$ as defined in (2.3):

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{F} \hat{\boldsymbol{x}}^{\otimes (q+2)} = -i \boldsymbol{H} \hat{\boldsymbol{x}}. \tag{2.13}$$

The Hamiltonian $\boldsymbol{H}$ is Hermitian and the entries of $\boldsymbol{F}$ must be real for a real system. Thus, all elements of $\boldsymbol{H}$ must be pure imaginary, so that the matrix $-i\boldsymbol{H}$ is an antisymmetric, real matrix. This antisymmetry manifests in the entries of $\boldsymbol{F}$, specifically in the first index and one other index, which we choose to be the second index. Thus, we require that $\boldsymbol{F}$ be antisymmetric in its first two indices. For a general tensor, this requirement can be enforced via (Engel 2023)

$$\boldsymbol{A}_\alpha = \frac{1}{q+3} \sum_{i=2}^{q+3} \left(\boldsymbol{F}_{P_{2i}\alpha} - \boldsymbol{F}_{P_{1i}P_{2i}\alpha} \right), \tag{2.14}$$

where α is a multi-index of $q+3$ indices, for example, $\alpha = (1, 2, 3, 4)$ for $q = 1$. Here, P_{ji} denotes the permutation operator which swaps indices j and i, so that $P_{23}(1, 2, 3, 4) = (1, 3, 2, 4)$. A derivation of (2.14) can be found in Appendix A. The antisymmetry of $\boldsymbol{A}$ allows the dynamics to be written in the Hamiltonian form.

As with the original tensor form, the antisymmetric tensor is not uniquely determined. Equation (2.14) results in one of many equivalent antisymmetric tensors for a given $\boldsymbol{F}$, owing to the redundancy in tensor forms. Enforcing antisymmetry in the first two indicies reduces, but does not eliminate, this redundancy.

The tensor yielded by (2.14) is not the only antisymmetric tensor which reproduces the desired dynamics. As with the general tensor form, the coefficient in $\boldsymbol{A}_\alpha$ may be partitioned across or combined with the entries indexed by a permutation on the latter indicies of α, as long as the antisymmetry is preserved. Further, although we

https://doi.org/10.1017/S0022377825000091 Published online by Cambridge University Press

have chosen $\boldsymbol{A}$ to be antisymmetric in its first two indices, in principle, a mapping to Hamiltonian form is possible as long as $\boldsymbol{A}$ is antisymmetric under the exchange of the first index and any of the latter indices. The choice of the second index is arbitrary.

2.6. *Mapping to standard form: enforcement of cubic degree*

The standard form (1.2) and (2.3) together yield a cubic system, with two powers of $|\psi\rangle$ coming from the expectation of the observable in (2.3) and the final power included in (1.2). To map a general dynamical system to the standard form, the degree-$(q+2)$ system encoded in the rank-$(q+3)$ tensor $\boldsymbol{A}$ must be reduced to a degree-3 system in a rank-4 tensor.

For odd q, we can apply the degree-reduction technique (Engel 2023), which evolves polynomials in $\hat{\boldsymbol{x}}$,

$$\hat{\boldsymbol{y}}_\alpha \propto \prod_{i=1}^{(q+1)/2} \hat{\boldsymbol{x}}_{\alpha_i}, \tag{2.15}$$

using a modified tensor form

$$\frac{d}{\mathrm{d}t'}\hat{\boldsymbol{y}} = \boldsymbol{M}\hat{\boldsymbol{y}}^{\otimes 3}, \tag{2.16}$$

which preserves the system's dynamics. The exact form of $\boldsymbol{M}$ is not uniquely determined, but one possible choice is

$$\boldsymbol{M}_{\alpha\beta\nu\eta} = \sum_{i=1}^{(q+1)/2} \Big[\prod_{j\neq i} \delta_{\alpha_j\beta_j}\Big] \boldsymbol{A}_{\alpha_i\beta_i\nu\eta}. \tag{2.17}$$

Here, the variables α, β, ν and η each denote a multi-index of $(q+1)/2$ indices. In the case of even q, an extra constant factor of x_0 can be used to raise the system to odd degree. Applying the degree-reduction technique to the antisymmetric tensor $\boldsymbol{A}$ yields a cubic system, which maps to (2.3). As an example, (3.5) gives the reduced tensor form of the logistic system.

2.7. *Determining observable Hamiltonian pairs*

Having found the rank-4 tensor $\boldsymbol{M}_{a\beta\mu\nu}$, which is antisymmetric in its first two indices, it is possible to extract the Hermitian observables and Hamiltonians appearing in (2.3). The real system has two equivalent forms:

$$\begin{aligned} \frac{d}{\mathrm{d}t'}y_\alpha &= \sum_{\beta,\nu,\eta} \boldsymbol{M}_{\alpha\beta\nu\eta} y_\beta y_\nu y_\eta, \\ \frac{d}{\mathrm{d}t'}y_\alpha &= -i \sum_{\beta,\nu,\eta} \Big(\sum_{j,k} \boldsymbol{O}^{(\nu\eta)}_{jk} y_j y_k \Big) \boldsymbol{H}^{(\nu\eta)}_{\alpha\beta} y_\beta. \end{aligned} \tag{2.18}$$

In the later form, raised indices denote a specific observable Hamiltonian pair, while lowered indices denote a position within a matrix.

We have ensured that the first two indices of the tensor $\boldsymbol{M}$ are antisymmetric, while the order of the latter pair of indices is arbitrary. This freedom allows us to

https://doi.org/10.1017/S0022377825000091 Published online by Cambridge University Press

chose $\boldsymbol{M}_{\alpha\beta\nu\eta}$ to be non-zero only if $\eta \geqslant \nu$. The remaining requirement is that the matrices $\boldsymbol{O}^{(\nu\eta)}$ and $\boldsymbol{H}^{(\nu\eta)}$ are Hermitian. A viable mapping from the first to the second form in (2.18) is

$$\begin{aligned} \boldsymbol{O}^{(\nu\eta)}_{jk} &= \frac{1}{2}\left(\delta_{\nu j}\delta_{\eta k} + \delta_{\nu k}\delta_{\eta j}\right), \\ \boldsymbol{H}^{(\nu\eta)}_{\alpha\beta} &= i\boldsymbol{M}_{\alpha\beta\nu\eta}. \end{aligned} \tag{2.19}$$

It is worth pointing out that (2.19) is not necessarily the optimal mapping that produces the least number of observable-Hamiltonian pairs. For example, as we shall see in § 3.1, this mapping requires two pairs for the logistic system, for which one pair would be sufficient, owing to the fact that the two pairs generated by this algorithm share a Hamiltonian, up to a constant. Although not optimal, (2.19) is applicable to general systems.

The proposed form of $\boldsymbol{O}^{(\nu\eta)}$ has the additional benefit of being simple to measure. In the case where $\nu = \eta$, the matrix form of $\boldsymbol{O}^{(\nu\nu)}$ is simply the zero matrix with a single 1 on the diagonal, and its expectation value is simply y_ν^2, which is thus the probability of measuring the νth state during projective measurement. In the case where $\nu \neq \eta$, $\boldsymbol{O}^{(\nu\eta)}$ is mostly zero but has two off-diagonal entries equal to $\frac{1}{2}$. In this case, the expectation value of $\boldsymbol{O}^{(\nu\eta)}$ equalling $y_\nu y_\eta$ can be found using projective measurement and a relative phase-finding algorithm (Shi *et al.*, 2024*a*).

2.8. *Convergence to the stochastic model*

Our approach uses measurements to achieve linearisation, which injects statistical fluctuations into otherwise deterministic systems. Evaluating (2.3) at the beginning of each time step requires finding $\langle\psi|\boldsymbol{O}_k|\psi\rangle$. At the beginning of each step, m measurements of $\boldsymbol{O}_k$ generate a sample mean for the expectation value $\langle\psi|\boldsymbol{O}_k|\psi\rangle$. As m increases to infinity, this sample mean converges to the exact value, but for finite m, the sample is a multinomial random variable, with non-zero variance in its mean. Variation in the measured values of $\langle\psi|\boldsymbol{O}_k|\psi\rangle$ causes spreading in simulated trajectories and divergence from the exact solution. For sufficiently large m and small Δt, the quantum algorithm converges to the solution of a stochastic differential equation, with the system's stochasticity quantified by the rate $s = m/\Delta t$ of measurements per unit time. As s increases to infinity, either by increasing the number of measurements or shortening time steps, the quantum simulation and its stochastic model converge to the exact solution. However, over a total simulation time T, the quantum algorithm requires MTs measurements. In this paper, we refer to a deterministic, $s \to \infty$ solution as the exact solution. We also present finite-s, quantum trajectories. Finding the $s \to \infty$ solution is only possible on classical computers.

2.9. *Quantifying stochasticity: ensembles and entropy*

Due to its stochastic nature, the algorithm produces a different trajectory each time, even for fixed systems, initial conditions and sampling rates. Taking the quantum states $|\psi\rangle$ of K trajectories as an ensemble, we can define a density matrix ρ for the states as

$$\rho = \frac{1}{K}\sum_{j=1}^{K} |\psi_j\rangle\langle\psi_j|. \tag{2.20}$$

https://doi.org/10.1017/S0022377825000091 Published online by Cambridge University Press

As the trajectories spread due to the algorithm's stochasticity, the von Neumann entropy S (Nielsen & Chuang 2010) of N qubits,

$$S = -\text{Trace}[\rho \log \rho] \leqslant N \log 2, \tag{2.21}$$

of this density matrix increases. At the beginning of the simulation, each of the trajectories has the exact same initial state and thus the density matrix describes a pure state, with zero entropy. For a deterministic simulation, the states of the ensemble would remain identical and the entropy does not increase. However, the stochasticity of the quantum simulation introduces small impurities to the ensemble, which become more pronounced as the simulation progresses. Thus, under a stochastic scheme, von Neumann entropy increases with the runtime.

In addition to measuring the intrinsic spread of an ensemble, we can also compare the quantum ensemble with a pure state, which is formed by sampling the quantum states in the deterministic trajectory on classical computers. The difference between two ensembles encoded in ρ_1 and ρ_2 can be quantified by the trace distance

$$T(\rho_1, \rho_2) = \frac{1}{2}||\rho_1 - \rho_2||_1 \leqslant 1. \tag{2.22}$$

Encoding the deterministic solution as the amplitudes of a quantum state, and then as a density matrix, allows us to use the trace distance as a measure of error between the quantum ensemble and deterministic solution.

2.10. *Algorithm scaling*

Having described how to map a general nonlinear system (1.1) to the standard cubic Hamiltonian form (2.3) that is amenable to quantum Hamiltonian simulation using piece-wise linearisation (2.2), let us briefly discuss the numerical cost of our approach. Measuring all M observables over a simulation time T, with m measurements per observable per time step, requires $\mathcal{O}(MmT/\Delta t)$ total measurements and each measurement destroys one copy of the current state $|\psi\rangle_{n-1}$. On average, each copy of the state will be evolved for half the total runtime, meaning the number of Hamiltonian simulation steps is $\mathcal{O}(MmT^2/2\Delta t^2)$. The total cost of the measurements and Hamiltonian simulation steps are then $\mathcal{O}(g(MmT)/(\Delta t))$ and $\mathcal{O}(h(MmT^2)/(2\Delta t^2))$, respectively, where g is the cost of a single measurement and h the cost of a single evolution step. In general, g and h depend on a number of factors. For example, certain observables are more easily measured than others and Hamiltonian simulation is most efficient for sparse matrices.

The scaling of the algorithm depends on the number of observable-Hamiltonian pairs, but also the exact nature of those pairs and whether the final total Hamiltonian can be evolved forward efficiently. For most systems, this algorithm is unlikely to be efficient, but there may exist specific systems, potentially those which can be realised with a few, sparse sub-Hamiltonians, for which the algorithm has a chance of outperforming existing quantum methods. However, more research is needed to confirm whether any systems of interest to plasma physics are of this form.

Of additional interest is the scaling of m with error and runtime. Although we previously assumed m to be chosen arbitrarily, we expect the algorithm's results to improve with increasing m and we might therefore define a critical $m^*(T, \epsilon)$ which, for a given system and runtime T, satisfies a maximum error bound ϵ. The scaling of m^* is system-dependent; for example, we expect m^* to scale poorly in chaotic systems. However, we can investigate the $M = 1$ case as a potential baseline.

https://doi.org/10.1017/S0022377825000091 Published online by Cambridge University Press

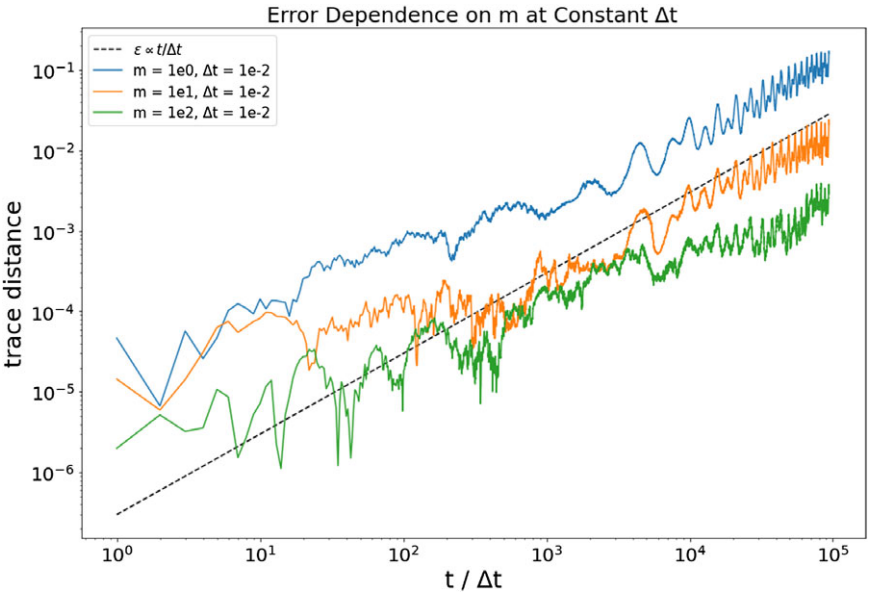


FIGURE 1. Trace distance between a deterministic solution and a stochastic, finite m ensemble of 500 independent trajectories; the error $\epsilon(t/\Delta t)$ scales with m^{-1} and is linear in time.

In figure 1, we plot the development of error for 500 trajectories in a system with a single, randomly generated observable-Hamiltonian pair and find error to be linear in time and inversely proportional to m. Thus, we would expect m^* for this system to be $\mathcal{O}(T/\epsilon)$ or potentially better following amplitude amplification (Brassard *et al.* 2002). While systems of this type are overly simple, they are potentially relevant; the logistic system (3.1) can be realised using a single pair (3.6).

Despite the linear scaling demonstrated for $M = 1$ in figure 1, we cannot expect sub-exponential scaling to be universal for more complex, $M > 1$ systems. For example, under chaotic dynamics, small perturbations between trajectories, including the finite-m stochastic fluctuations, tend to grow exponentially. This fact is a fundamental limit on quantum algorithms for nonlinear dynamics (Lewis *et al.* 2024).

3. Algorithm demonstrations

To demonstrate how to apply our approach to a general polynomial nonlinear system, we use the logistic system and the Lorenz system as two examples. The logistic system is used as a simple, easily verifiable test case, and the Lorenz system is used as a chaotic model of dissipative fluid mechanics and thus as a precursor to plasma dynamics.

3.1. *Solutions to the logistic system*

The logistic system is commonly used to describe populations and other systems which grow to a maximum value (Verhulst 1838). In normalised units, the logistic system is described by the quadratic ODE:

$$\dot{x}_1 = x_1(1 - x_1). \tag{3.1}$$

https://doi.org/10.1017/S0022377825000091 Published online by Cambridge University Press

While this system is relatively simple, previous quantum algorithms have not always successfully reproduced the dynamics (Sanavio *et al.* 2024). Here, we use the logistic system as a simple demonstration of our approach. For this system, the mapping is simple enough to be completed by hand. As a first step, we introduce the constant $x_0 = 1$ to this system,

$$\begin{aligned} \dot{x}_0 &= 0, \\ \dot{x}_1 &= x_0^2 x_1 - x_0 x_1^2, \end{aligned} \tag{3.2}$$

to raise it to an homogeneous degree-3 system.

Using (2.10), without the prefactor, we generate the norm-preserving system:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t'}\hat{x}_0 &= +\hat{x}_0^2\hat{x}_1^3 - \hat{x}_0^3\hat{x}_1^2 = \boldsymbol{A}_{010011}\hat{x}_1\hat{x}_0\hat{x}_0\hat{x}_1\hat{x}_1 + \boldsymbol{A}_{010001}\hat{x}_1\hat{x}_0\hat{x}_0\hat{x}_0\hat{x}_1, \\ \frac{\mathrm{d}}{\mathrm{d}t'}\hat{x}_1 &= -\hat{x}_0^3\hat{x}_1^2 + \hat{x}_0^4\hat{x}_1^1 = \boldsymbol{A}_{100011}\hat{x}_0\hat{x}_0\hat{x}_0\hat{x}_1\hat{x}_1 + \boldsymbol{A}_{100001}\hat{x}_0\hat{x}_0\hat{x}_0\hat{x}_0\hat{x}_1, \end{aligned} \tag{3.3}$$

corresponding to four non-zero elements of $\boldsymbol{A}$:

$$\begin{aligned} \boldsymbol{A}_{100001} &= -\boldsymbol{A}_{010001} = 1, \\ \boldsymbol{A}_{010011} &= -\boldsymbol{A}_{100011} = 1. \end{aligned} \tag{3.4}$$

For this simple system, we can find $\boldsymbol{M}$ manually:

$$\begin{aligned} \boldsymbol{M}_{(10)(00)\mu\nu} = \boldsymbol{M}_{(11)(01)\mu\nu} = \boldsymbol{M}_{(01)(00)\mu\nu} = \boldsymbol{M}_{(11)(10)\mu\nu} = \boldsymbol{A}_{(1)(0)\mu\nu} = +1, \\ \boldsymbol{M}_{(00)(10)\mu\nu} = \boldsymbol{M}_{(01)(11)\mu\nu} = \boldsymbol{M}_{(00)(01)\mu\nu} = \boldsymbol{M}_{(10)(11)\mu\nu} = \boldsymbol{A}_{(0)(1)\mu\nu} = -1, \\ \boldsymbol{M}_{(00)(10)\mu\xi} = \boldsymbol{M}_{(01)(11)\mu\xi} = \boldsymbol{M}_{(00)(01)\mu\xi} = \boldsymbol{M}_{(10)(11)\mu\xi} = \boldsymbol{A}_{(0)(1)\mu\xi} = +1, \\ \boldsymbol{M}_{(10)(00)\mu\xi} = \boldsymbol{M}_{(11)(01)\mu\xi} = \boldsymbol{M}_{(01)(00)\mu\xi} = \boldsymbol{M}_{(11)(10)\mu\xi} = \boldsymbol{A}_{(1)(0)\mu\xi} = -1, \end{aligned} \tag{3.5}$$

where $\mu = (00)$, $\nu = (01)$ and $\xi = (11)$. In (3.5), parentheses have been added to the multi-indices of $\boldsymbol{A}$ to highlight the relationship with the multi-indices of $\boldsymbol{M}$; these parentheses have no significance other than indicating the grouping of indices. We then map this result to the observable-Hamiltonian form accepted by the algorithm:

$$\boldsymbol{O} = \begin{bmatrix} 0 & \frac{1}{2} & 0 & -\frac{1}{2} \\ \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 \end{bmatrix}; \quad \boldsymbol{H} = \begin{bmatrix} 0 & -i & -i & 0 \\ i & 0 & 0 & -i \\ i & 0 & 0 & -i \\ 0 & i & i & 0 \end{bmatrix}. \tag{3.6}$$

After encoding the system into the quantum amplitudes and evolving according to our algorithm, we recover the final normalisation of the trajectory by finding the value of $\hat{x}_0$, using this to determine the final value of $|\boldsymbol{x}|$ and thus recovering x_1. The results of the algorithm on a classical simulator are shown figure 2. For simple and convergent dynamics like the logistic system, the underlying behaviour dominates over any stochastic fluctuations, resulting in minimal spreading. Under these conditions, the quantum trajectories closely follow the deterministic solution.

https://doi.org/10.1017/S0022377825000091 Published online by Cambridge University Press

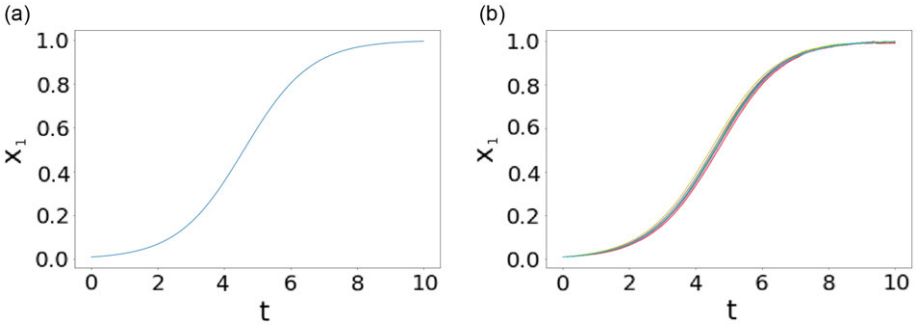


FIGURE 2. Time evolution of the logistic system with initial condition $x_1(0) = 10^{-2}$. The deterministic trajectory in panel (*a*) can be recovered as the average of a collection of 10 quantum trajectories in panel (*b*), which use the stochastic parameter $s = 5 \times 10^5$.

3.2. *Solutions to the Lorenz system*

As the next non-trivial example, we apply our approach to the Lorenz system (Lorenz 1963):

$$\begin{aligned} \dot{x}_1 &= \sigma(x_2 - x_1), \\ \dot{x}_2 &= x_1(\rho - x_3) - x_2, \\ \dot{x}_3 &= x_1 x_2 - \beta x_3. \end{aligned} \tag{3.7}$$

Originally used to describe atmospheric flow, the Lorenz system is a classic example of chaotic dynamics, although the system is only chaotic under certain parameters (Hirsch, Smale & Devaney 2003): $\beta \approx 8/3$, $\rho \approx 28$, $\sigma \approx 10$, and away from this region, the flow is integrable. The system generally has multiple fixed points, at the origin and the points

$$\left(\pm \sqrt{\beta(\rho - 1)}, \pm \sqrt{\beta(\rho - 1)}, \rho - 1 \right).$$

Outside the chaotic regime, these fixed points are attractors, with the system's overall behaviour highly dependent on the behaviour around these points. The chaotic system follows a butterfly-shaped trajectory around two of these points, with the solution switching between the two unpredictably. The inherent chaos makes the Lorenz system a useful bridge to plasma physics, which may also encounter chaotic dynamics, for example, with turbulent flows.

The homogeneous Lorenz system is degree-2, but it is necessary to add an extra constant factor to raise it to degree-3 for later degree reduction, which requires odd degrees. For the cubic system, the norm-preserving system in (2.10) is degree-5, per (2.12). By mapping to quadratic polynomials in the entries of $\hat{\boldsymbol{x}}$, we reduce the system to degree-3 on 16 dynamic variables. The 16-variable state of the system can be stored on 4 qubits and the final mapping requires at most 26 observable-Hamiltonian pairs. This mapping may not be optimal. We automate the mapping protocol using an openly available Python code (Andress 2024).

We run our algorithm for example problems. First, for a well-behaved Lorenz system (figure 3*a,b*), the trajectories remain together for the entire simulation. By the time that noticeable spread would occur, the system has converged to its steady-state solution. Second, for a chaotic Lorenz system (figure 3*c,d*), stochastic spread

https://doi.org/10.1017/S0022377825000091 Published online by Cambridge University Press

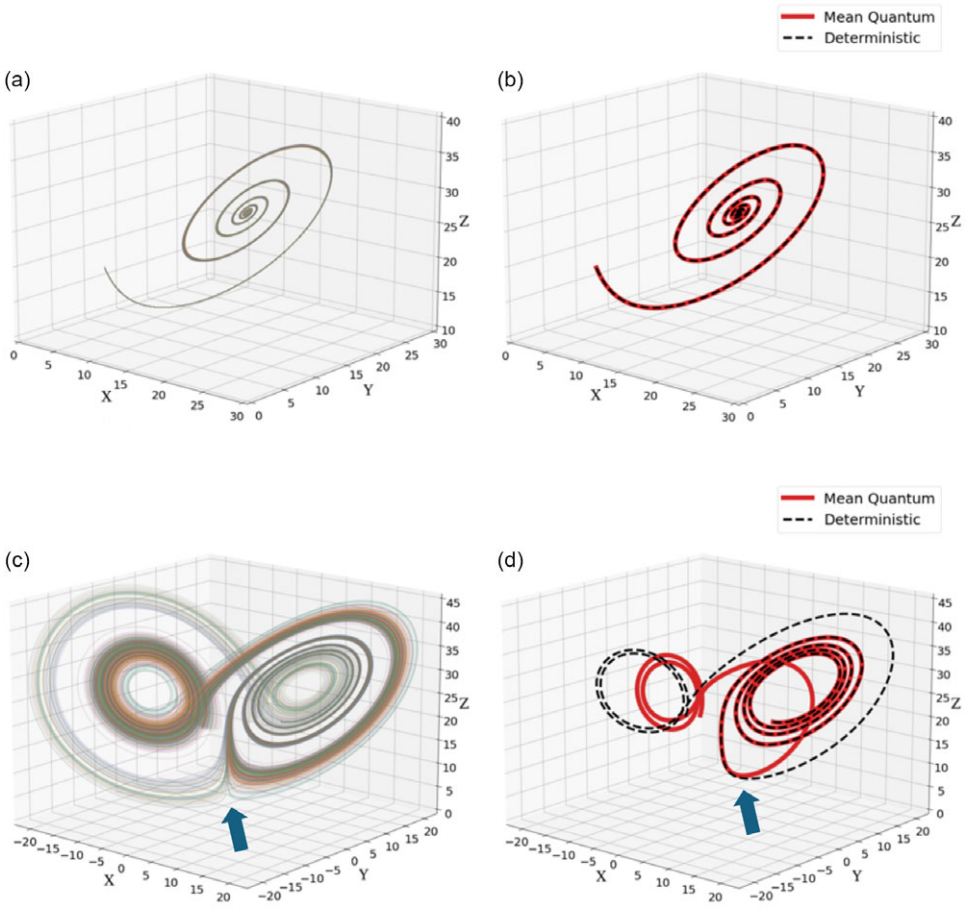


FIGURE 3. Time evolution of the (*a,b*) well-behaved and (*c,d*) chaotic Lorenz system with runtime $T = 5$ and parameters $\rho = 28$, $\sigma = 10$, $\beta = 10$ (well-behaved), $\beta = 8/3$ (chaotic) and initial conditions $\boldsymbol{x} = (4.856, 7.291, 18.987)$. (*a,c*) A collection of 300 independent, stochastic quantum trajectories with $\Delta t = 10^{-5}$, $s = 10^{15}$ in panels (*a,c*) yield a mean trajectory (red) which initially follows the $\Delta t = 10^{-5}$, $s \to \infty$ in panels (*b,d*), deterministic solution (black). At the point indicated by arrows in panels (*c,d*), the chaotic trajectories diverge, resulting in eventual deviation from the deterministic solution.

in the trajectories causes significant divergence at later times. Of primary interest is the branching point indicated by the arrow. This branching point corresponds to the first sizeable divergence between the mean trajectory and deterministic solution, as shown in figure 3(*d*).

3.3. *Error and entropy of the chaotic Lorenz system*

Taking the trajectories in figure 3(*c*) as an ensemble, the increase in trace distance to the deterministic solution is shown in figure 4. As demonstrated in the inset, the von Neumann entropy and the trace distance (error) between the ensemble and deterministic trajectories are strongly correlated. The arrow in figure 4(*a*) corresponds to the branching point in figure 3(*c,d*). As a comparison, no such corner

https://doi.org/10.1017/S0022377825000091 Published online by Cambridge University Press

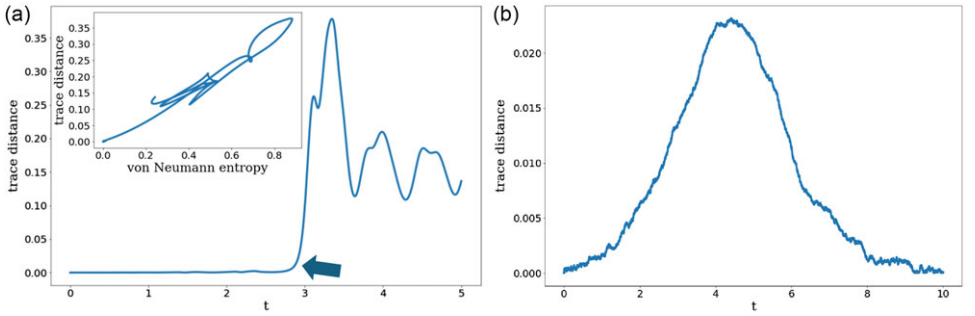


FIGURE 4. (*a*) Trace distance between the ensemble of trajectories in figure 3(*c*) and the deterministic solution increases suddenly at the indicated branching point, and closely correlates to the von Neumann entropy (inset). (*b*) Trace distance between the ensemble of trajectories in figure 2(*a*) remains low throughout the simulation and returns to near-zero as the system converges.

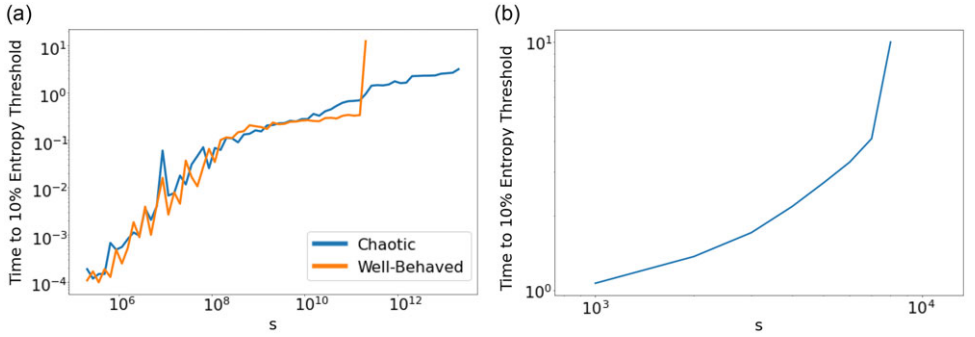


FIGURE 5. Timing of the first branch of the quantum algorithm's results, signalled by an increase to 10 % of the maximum entropy, as a function of s for the (*a*) Lorenz and (*b*) logistic systems; at low s, the integrable (orange) and chaotic (blue) Lorenz systems scale similarly, but past a threshhold at $s \sim 10^{11}$, the integrable solutions converge and no longer branch. This phenomenon does not appear for the chaotic system, in which branching is inevitable, but does occur in the logistic system.

exists in the error data for the logistic system (figure 4*b*) and the error decreases dramatically as the system converges.

Beyond the point of divergence, the quantum algorithm is no longer reliable, so if we desire reliable results for a set runtime T, we must find some method to delay the branching point. The most obvious method is to increase the stochastic sampling rate s; as s increases, the algorithm should remain reliable longer. However, because the algorithm requires MTs measurements, an incentive exists to minimise s for a given runtime T. Figure 5(*a*) shows the time of divergence as a function of s. For the well-behaved system and the logistic system (figure 5*b*), a threshold exists beyond which entropy never surpasses 10 % of its maximum value, because all solutions reach the fixed point. In other words, for the well-behaved system, there exists an upper bound for s beyond which, error stays below a threshold for all time. In comparison, for the chaotic system, error always exceeds a given value after some finite time. Thus, for

https://doi.org/10.1017/S0022377825000091 Published online by Cambridge University Press

the chaotic system, branching is inevitable, but its onset can be delayed by increasing s. For the algorithm to be efficient, the dynamical system must have the property that s, the requisite stochastic sampling rate, increases slowly with T, the targeted runtime.

4. Summary

We have developed a method to map any real, polynomial dynamical system to a form which could be implemented on future fault-tolerant quantum hardware. This mapping results in a cubic, norm-preserving form which can be stored in a quantum state and evolves via a nonlinear Hamiltonian. We defined the nonlinear Hamiltonian as the sum of constant Hamiltonians, weighted by quadratic observables, and we approximate the nonlinear Hamiltonian as being constant over a small time period, allowing for the use of Hamiltonian simulation methods. By applying the algorithm to the logistic and Lorenz systems, we demonstrated that this map recovers the classical dynamics and can detect the onset of chaos. In the non-chaotic regime of the Lorenz system, a minimum value of the stochastic sampling rate was found to ensure that the simulated and deterministic solutions never diverge, but no threshold was observed in the chaotic regime.

Of future interest are systems which are neither convergent nor chaotic. Under logistic dynamics and the well-behaved Lorenz system, early convergence to equilibrium prevents a deeper study of the algorithm's long-term error. Meanwhile, under chaotic dynamics, even classical algorithms are expected to diverge quickly. More study is required between these two extremes to determine long-term behaviour for non-trivial, non-chaotic dynamics.

Whether this algorithm can be implemented efficiently on quantum hardware remains an open question. For systems that can only be realised with a large number of observables, or that include dense sub-Hamiltonians, it is unlikely that an algorithm of this form could be efficient. More research is needed to further investigate the efficiency of a measurement-based approach. Nevertheless, variants of the algorithm presented here may solve classes of problems not covered by existing quantum algorithms.

Funding

This work was supported by the U.S. Department of Energy under Grant No. DE-SC0020393.

Competing interests

The authors declare none.

Code availability

The code developed for this paper is available from Andress (2024).

Appendix A.

A.1. *Derivation of norm-preserving dynamics*

Following Engel (2023), (2.10) may be derived using the product rule for derivatives:

https://doi.org/10.1017/S0022377825000091 Published online by Cambridge University Press

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\boldsymbol{x}}{|\boldsymbol{x}|}\right) &= \frac{1}{|\boldsymbol{x}|}\frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{x}) + \boldsymbol{x}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{|\boldsymbol{x}|}\right) \\ &= |\boldsymbol{x}|^{-1}\boldsymbol{G}(\boldsymbol{x}) - |\boldsymbol{x}|^{-3}[\boldsymbol{x}\cdot\boldsymbol{G}(\boldsymbol{x})]\boldsymbol{x},\end{aligned} \tag{A.1}$$

and using (2.9), when the system is homogeneous, we can pull $|\boldsymbol{x}|$ to the front:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\boldsymbol{x}}{|\boldsymbol{x}|}\right) = |\boldsymbol{x}|^{q-1}\left(\boldsymbol{G}(\hat{\boldsymbol{x}}) - [\hat{\boldsymbol{x}}\cdot\boldsymbol{G}(\hat{\boldsymbol{x}})]\hat{\boldsymbol{x}}\right). \tag{A.2}$$

Inserting a factor of $|\hat{\boldsymbol{x}}|^2$ to the first term to yield (2.10) does not affect the system's dynamics because, by definition, $|\hat{\boldsymbol{x}}|$ is always one. The insertion of that factor ensures that the final expression is an homogeneous polynomial.

A.2. *Derivation of anti-symmetric forms*

The derivation of (2.14) requires manipulation of the norm-preserving tensors. For a tensor system, such as in (2.12), to be norm-preserving requires that

$$\hat{\boldsymbol{x}}\cdot\frac{\mathrm{d}}{\mathrm{d}t'}\hat{\boldsymbol{x}} = \boldsymbol{F}x^{\otimes(q+3)} = 0, \tag{A.3}$$

containing an implicit sum of products in the entries of $\boldsymbol{x}$, and by the commutativity of multiplication, a given set of factors in $\boldsymbol{x}$ may correspond to more than one entry in $\boldsymbol{F}$ (Engel 2023). However, because the entries in $\boldsymbol{x}$ are independent of each other, the only way for (A.3) to be identically zero is for entries to be cancelled. Thus, for a multi-index α of $q+3$ indices,

$$\sum_{P\in S_{q+3}} \boldsymbol{F}_{P\alpha} = 0, \tag{A.4}$$

where S_{q+3} is the set of permutations on $q+3$ elements (Engel 2023).

Because the dynamical system must eventually be mapped to a Hamiltonian, it is necessary to enforce a certain symmetry on the tensor form. Hamiltonians are Hermitian, but because Schrödinger's equation (1.2) includes an extra imaginary factor, the matrices encoded in the tensor system must be anti-Hermitian. Hence, the tensor system must be antisymmetric in its first two indices. For a given, norm-preserving dynamical system, the form of the tensor $\boldsymbol{F}$ is not unique and thus we find a method of converting a general $\boldsymbol{F}$ to this antisymmetric form. To enforce antisymmetry onto $\boldsymbol{F}$ without changing the system's dynamics (Engel 2023) takes

$$\begin{aligned}\tilde{\boldsymbol{F}}_\alpha &\equiv \frac{1}{q+3}\left(2(q+2)\boldsymbol{F}_\alpha - (q+1)\boldsymbol{F}_\alpha + \sum_{i=2}^{q+3}\boldsymbol{F}_{P_{1i}\alpha} - \sum_{i=2}^{q+3}\boldsymbol{F}_{P_{1i}\alpha}\right) \\ &\equiv \frac{1}{q+3}\left[\sum_{i=2}^{q+3}\left(\boldsymbol{F}_\alpha - \boldsymbol{F}_{P_{1i}\alpha}\right) + \sum_{i=2}^{q+3}\left(\boldsymbol{F}_\alpha + \boldsymbol{F}_{P_{1i}\alpha}\right) - (q+1)\boldsymbol{F}_\alpha\right].\end{aligned} \tag{A.5}$$

Again, by the commutativity of multiplication, the indices of $\boldsymbol{F}$, with the exception of the first index, can be reordered without affecting the system's dynamics. Thus, (A.5) is equivalent to the sum of $(1/q+3)\sum_{i=2}^{q+3}(\boldsymbol{F}_{P_{2i}\alpha} - \boldsymbol{F}_{P_{1i}P_{2i}\alpha})$ and $(1/q+3)[\boldsymbol{F}_\alpha + \sum_{i=2}^{q+3}\boldsymbol{F}_{P_{1i}\alpha}]$ (Engel 2023).

https://doi.org/10.1017/S0022377825000091 Published online by Cambridge University Press

The second term $(1/q+3)[\boldsymbol{F}_\alpha + \sum_{i=2}^{q+3} \boldsymbol{F}_{P_{1i}\alpha}]$ reduces to $(1/q+3)\boldsymbol{T}_\alpha \equiv \frac{1}{q+3}\sum_{i=1}^{q+3} \boldsymbol{F}_{P_{1i}\alpha}$. Notably, because the order of the final $q+2$ indices are arbitrary, we can replace $\boldsymbol{T}_{j\beta}$ with

$$\tilde{\boldsymbol{T}}_{j\beta} \equiv \frac{1}{(q+2)!}\sum_{P\in S_{q+2}} \boldsymbol{T}_{jP\beta} \tag{A.6}$$

for any multi-index β of $q+2$ indices; the effect of (A.6) is to enfore symmetry in $\boldsymbol{T}$ across the latter indices. We see then

$$\frac{1}{(q+2)!}\sum_{P'\in S_{q+2}} \boldsymbol{T}_{jP\beta} = \frac{1}{(q+2)!}\sum_{P''\in S_{q+3}} \boldsymbol{F}_{P''[j\beta]};$$

because the permutation P' shuffles the latter indices and the P_{1i} swaps the first index into any of the latter positions, all possible ordering of the multi-index $j\beta$ are reached. Thus, by (A.4), the second term in the sum for $\tilde{\boldsymbol{F}}_\alpha$ must be zero, yielding (2.14):

$$\tilde{\boldsymbol{F}}_\alpha \equiv \frac{1}{q+3}\sum_{i=2}^{q+3}\left(\boldsymbol{F}_{P_{2i}\alpha} - \boldsymbol{F}_{P_{1i}P_{2i}\alpha}\right).$$

The two sides of (2.14) yield equivalent dynamics, but the right side is in a form which is antisymmetric in the first two indices.

A.2.1. A simple example

To demonstrate how (2.14) can be applied, let us first consider the following norm-preserving system:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t'}\hat{x}_1 &= \hat{x}_2\hat{x}_3,\\
\frac{\mathrm{d}}{\mathrm{d}t'}\hat{x}_2 &= -\hat{x}_1\hat{x}_3,\\
\frac{\mathrm{d}}{\mathrm{d}t'}\hat{x}_3 &= 0.
\end{aligned} \tag{A.7}$$

The tensor system $\boldsymbol{F}_{123}=1$, $\boldsymbol{F}_{213}=-(1/2)$, $\boldsymbol{F}_{231}=-(1/2)$ realises the dynamics in (A.7). Following through (2.14) yields the non-zero elements of an equivalent, antisymmetric tensor:

$$\begin{aligned}
\boldsymbol{A}_{123} &= \frac{1}{3}\Big[(\boldsymbol{F}_{123}-\boldsymbol{F}_{213})+(\boldsymbol{F}_{132}-\boldsymbol{F}_{231})\Big] = +\frac{2}{3},\\
\boldsymbol{A}_{132} &= \frac{1}{3}\Big[(\boldsymbol{F}_{132}-\boldsymbol{F}_{312})+(\boldsymbol{F}_{123}-\boldsymbol{F}_{321})\Big] = +\frac{1}{3},\\
\boldsymbol{A}_{213} &= \frac{1}{3}\Big[(\boldsymbol{F}_{213}-\boldsymbol{F}_{123})+(\boldsymbol{F}_{231}-\boldsymbol{F}_{132})\Big] = -\frac{2}{3},\\
\boldsymbol{A}_{231} &= \frac{1}{3}\Big[(\boldsymbol{F}_{231}-\boldsymbol{F}_{321})+(\boldsymbol{F}_{213}-\boldsymbol{F}_{312})\Big] = -\frac{1}{3},
\end{aligned}$$

https://doi.org/10.1017/S0022377825000091 Published online by Cambridge University Press

$$\boldsymbol{A}_{312} = \frac{1}{3}\Big[(\boldsymbol{F}_{312} - \boldsymbol{F}_{132}) + (\boldsymbol{F}_{321} - \boldsymbol{F}_{123})\Big] = -\frac{1}{3},$$
$$\boldsymbol{A}_{321} = \frac{1}{3}\Big[(\boldsymbol{F}_{321} - \boldsymbol{F}_{231}) + (\boldsymbol{F}_{312} - \boldsymbol{F}_{213})\Big] = +\frac{1}{3}. \tag{A.8}$$

Notably, this tensor has more non-zero elements than necessary. An alternative with the minimum number of non-zero elements is $\boldsymbol{A}_{123} = +1$, $\boldsymbol{A}_{213} = -1$. Tensor forms can be made more sparse by restricting the order of latter indices, namely the second through last indices of the input tensor $\boldsymbol{F}$ and the third through last of $\boldsymbol{A}$. Although not optimal, the tensor in (A.8) can be mapped to a Hamiltonian.

A.3. *Derivation of degree-reduced system*

Equation (2.17) is one implementation of a system which satisfies fundamental calculus rules for a product, because (Engel 2023)

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t'}\hat{\boldsymbol{y}}_\alpha &= \frac{\mathrm{d}}{\mathrm{d}t'}\prod_{i=1}^{(q+1)/2}\hat{\boldsymbol{x}}_{\alpha_i} \\ &= \sum_{i=1}^{(q+1)/2}\Big(\prod_{j\neq i}\hat{\boldsymbol{x}}_{\alpha_j}\Big)\frac{\mathrm{d}}{\mathrm{d}t'}\hat{\boldsymbol{x}}_{\alpha_i} \\ &= \sum_{i=1}^{(q+1)/2}\Big(\prod_{j\neq i}\hat{\boldsymbol{x}}_{\alpha_j}\Big)\Big(\sum_{\beta}\boldsymbol{A}_{\alpha_i\beta}\prod_{k=1}^{q+2}\hat{x}_{\beta_k}\Big),\end{aligned} \tag{A.9}$$

and using (2.17), we find

$$\begin{aligned}(\boldsymbol{M}\hat{\boldsymbol{y}}^{\otimes 3})_\alpha &= \sum_{\gamma,\nu,\eta}\boldsymbol{M}_{\alpha\gamma\nu\eta}\hat{\boldsymbol{y}}_\gamma\hat{\boldsymbol{y}}_\nu\hat{\boldsymbol{y}}_\eta \\ &= \sum_{\gamma,\nu,\eta}\left[\sum_{i=1}^{(q+1)/2}\Big[\prod_{j\neq i}\delta_{\alpha_j\gamma_j}\Big]\boldsymbol{A}_{\alpha_i\gamma_i\nu\eta}\right]\left(\prod_{k=1}^{(q+1)/2}\hat{\boldsymbol{x}}_{\gamma_k}\right)\hat{\boldsymbol{y}}_\nu\hat{\boldsymbol{y}}_\eta \\ &= \sum_{\gamma,\nu,\eta}\left[\sum_{i=1}^{(q+1)/2}\Big[\prod_{j\neq i}\hat{\boldsymbol{x}}_{\alpha_j}\Big]\boldsymbol{A}_{\alpha_i\gamma_i\nu\eta}\right](\hat{\boldsymbol{x}}_{\gamma_i})\hat{\boldsymbol{y}}_\nu\hat{\boldsymbol{y}}_\eta \\ &= \sum_{i=1}^{(q+1)/2}\Big(\prod_{j\neq i}\hat{\boldsymbol{x}}_{\alpha_j}\Big)\Big(\sum_{\gamma_i,\nu,\eta}\boldsymbol{A}_{\alpha_i\gamma_i\nu\eta}\hat{\boldsymbol{x}}_{\gamma_i}\hat{\boldsymbol{y}}_\nu\hat{\boldsymbol{y}}_\eta\Big) \\ &= \sum_{i=1}^{(q+1)/2}\Big(\prod_{j\neq i}\hat{\boldsymbol{x}}_{\alpha_j}\Big)\left(\sum_{\beta}\boldsymbol{A}_{\alpha_i\beta}\prod_{k=1}^{q+2}\hat{x}_{\beta_k}\right).\end{aligned} \tag{A.10}$$

Under the mapping equation (2.17), entries $\boldsymbol{M}_{\alpha\beta\eta\nu}$ must be zero if α and β differ in more than one index, and $\boldsymbol{M}_{\alpha\beta\eta\nu} = -\boldsymbol{M}_{\beta\alpha\eta\nu}$. Further, each non-zero entry of $\boldsymbol{M}$ is exactly equal to a single corresponding entry of $\boldsymbol{A}$, so a valid $\boldsymbol{M}$ can be reverse-mapped back to its $\boldsymbol{A}$.

https://doi.org/10.1017/S0022377825000091 Published online by Cambridge University Press

A.3.1. A simple example

To demonstrate the use of (2.17), let us consider the following antisymmetric, norm-preserving tensor system:

$$\begin{aligned} \boldsymbol{A}_{121111} &= -1, \\ \boldsymbol{A}_{211111} &= +1. \end{aligned} \tag{A.11}$$

For a rank-6 $\boldsymbol{A}$, the corresponding $\boldsymbol{M}$ must have four multi-indices of two indices each. In the case of only two variables, as in (A.11), this corresponds to a $4 \times 4 \times 4 \times 4$ $\boldsymbol{M}$ tensor.

For a sparse $\boldsymbol{A}$, as in (A.11), (2.17) determines which elements of $\boldsymbol{M}$ are non-zero. First, the indices of $\boldsymbol{A}$ are split into four multi-indices, with the first and second indices being part of α and β, respectively, the third and fourth indices being ν, and the fifth and sixth indices being η. The full set of multi-indices of $\boldsymbol{M}$ are determined then by filling out α and β with another index to create multi-indices of two indices. By the product of Kronecker deltas, these added indices must be identical and added in the same position of the multi-index. For example, the element $\boldsymbol{A}_{(1)(2)(11)(11)}$ would contribute to the elements $\boldsymbol{M}_{(11)(21)(11)(11)}$ and $\boldsymbol{M}_{(21)(22)(11)(11)}$, but not $\boldsymbol{M}_{(11)(22)(11)(11)}$ or $\boldsymbol{M}_{(12)(12)(11)(11)}$. Thus, each of the elements in (A.11) corresponds to four elements in the final $\boldsymbol{M}$, corresponding to adding the index 1 (2) to the first (second) position in α and β; these four elements may not be unique. The eight non-zero elements of $\boldsymbol{M}$ in the system are then

$$\begin{aligned} \boldsymbol{M}_{(11)(12)\nu\eta} = \boldsymbol{M}_{(11)(21)\nu\eta} = \boldsymbol{M}_{(21)(22)\nu\eta} = \boldsymbol{M}_{(12)(22)\nu\eta} = \boldsymbol{A}_{(1)(2)\nu\eta} = -1, \\ \boldsymbol{M}_{(12)(11)\nu\eta} = \boldsymbol{M}_{(21)(11)\nu\eta} = \boldsymbol{M}_{(22)(21)\nu\eta} = \boldsymbol{M}_{(22)(12)\nu\eta} = \boldsymbol{A}_{(2)(1)\nu\eta} = +1, \end{aligned} \tag{A.12}$$

where $\nu = \eta = (11)$.

REFERENCES

ANDRESS, J. 2024 Quantum simulation of nonlinear dynamical systems utilizing repeated measurement. Available at https://gitlab.com/joan1465/quantum-simulation-of-nonlinear-dynamical-systems-utilizing-repeated-measurement

BERRY, D. W., CHILDS, A. M., CLEVE, R., KOTHARI, R. & SOMMA, R. D. 2014 Exponential improvement in precision for simulating sparse hamiltonians. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, pp. 283–292. Association for Computing Machinery.

BERRY, D.W., CHILDS, A.M. & KOTHARI, R. 2015 Hamiltonian simulation with nearly optimal dependence on all parameters. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pp. 792–809. IEEE Computer Society.

BRASSARD, G., HØYER, P., MOSCA, M. & TAPP, A. 2002 Quantum amplitude amplification and estimation. *Quantum Computation and Information*, pp 53–74. American Mathematical Society.

CHILDS, A. & WIEBE, N. 2012 Hamiltonian simulation using linear combinations of unitary operations. *Quant. Inform. Comput.* **12** (11&12), 901–924.

DODIN, I. & STARTSEV, E. 2021 On applications of quantum computing to plasma simulations. *Phys. Plasmas* **28** (9), 092101.

ENGEL, A. 2023 Developing quantum algorithms for efficient classical physics simulations. *PhD thesis*, University of Colorado Boulder, Boulder, CO, USA.

ENGEL, A., SMITH, G. & PARKER, S. 2021 Linear embedding of nonlinear dynamical systems and prospects for efficient quantum algorithms. *Phys. Plasmas* **28** (6), 062305.

ENGEL, A., SMITH, G. & PARKER, S.E. 2019 Quantum algorithm for the Vlasov equation. *Phys. Rev. A* **100** (6), 062315.

HIRSCH, M., SMALE, S. & DEVANEY, R. 2003 *Differential Equations, Dynamical Systems, & An Introduction to Chaos*. Academic Press.

https://doi.org/10.1017/S0022377825000091 Published online by Cambridge University Press

JOSEPH, I. 2020 Koopman–von neumann approach to quantum simulation of nonlinear classical dynamics. *Phys. Rev. Res.* **2** (4), 043102.

JOSEPH, I., SHI, Y., PORTER, M., CASTELLI, A., GEYKO, V., GRAZIANI, F., LIBBY, S. & DUBOIS, J. 2023 Quantum computing for fusion energy science applications. *Phys. Plasmas* **30** (1), 010501.

KABIN, K. 2021 Adiabatic invariant of a charged particle moving in a magnetic field with a constant gradient. *Phys. Plasmas* **28** (12), 122101.

LEWIS, D., EIDENBENZ, S., NADIGA, B. & SUBAŞI, Y. 2024 Limitations for quantum algorithms to solve turbulent and chaotic systems. *Quantum* **8**, 1509.

LEYTON, S. K. & OSBORNE, T. J. 2008 A quantum algorithm to solve nonlinear differential equations. https://arxiv.org/abs/0812.4423

LIU, J.-P., KOLDEN, H. Ø., KROVI, H. K., LOUREIRO, N. F., TRIVISA, K. & CHILDS, A. M. 2021 Efficient quantum algorithm for dissipative nonlinear differential equations, *Proc. Natl Acad. Sci. USA* **118** (35), e2026805118.

LLOYD, S., DE PALMA, G., GOKLER, C., KIANI, B., LIU, Z.-W., M., MARVIAN, TENNIE, F. & PALMER, T. 2020 Quantum algorithm for nonlinear differential equations. https://arxiv.org/abs/2011.06571

LORENZ, E. 1963 Deterministic nonperiodic flow. *J. Atmos. Sci.* **20** (2), 130–141.

LUBASCH, M., JOO, J., MOINIER, P., KIFFNER, M. & JAKSCH, D. 2020 Variational quantum algorithms for nonlinear problems. *Phys. Rev. A* **101** (1), 010301.

MCLACHLAN, R. I. 2003 Spatial discretization of partial differential equations with integrals. *IMA J. Numer. Anal.* **23** (4), 645–664.

NIELSEN, M. & CHUANG, I. 2010 *Quantum COmputation and Quantum Information*. Cambridge University Press.

SANAVIO, C., SCATAMACCHIA, R., DE FALCO, C. & SUCCI, S. 2024 Three Carleman routes to the quantum simulation of classical fluids. *Phys. Fluids* **36** (5), 057143.

SHI, Y., BECK, K.M., KRUSE, V.A. & LIBBY, S.B. 2024*a* Preparing angular momentum eigenstates using engineered quantum walks. *Phys. Rev. A* **110** (6), 062214.

SHI, Y., *et al.*et al. 2021 Simulating non-native cubic interactions on noisy quantum machines. *Phys. Rev. A* **103** (6), 062608.

SHI, Y., *et al.*et al. 2024*b* Simulating nonlinear optical processes on a superconducting quantum device. *J. Plasma Phys.* **90** (6), 805900602.

SUZUKI, M. 1991 General theory of fractal path integrals with applications to many-body theories and statistical physics. *J. Math. Phys.* **32** (2), 400–407.

VERHULST, P.-F. 1838 Notice sur la loi que la population suit dans son accroissement. *Correspondence Mathematique et Physique* **10**, 113–129.

XUE, C., WU, Y.-C. & GUO, G.-P. 2021 Quantum homotopy perturbation method for nonlinear dissipative ordinary differential equations. *New J. Phys.* **23** (12), 123035.

YEPZ, J. 2001 Type-ii quantum computers. *Intl J. Mod. Phys. C* **12** (09), 1273–1284.

https://doi.org/10.1017/S0022377825000091 Published online by Cambridge University Press