

<https://doi.org/10.1038/s41534-025-01084-z>

# Further improving quantum algorithms for nonlinear differential equations via higher-order methods and rescaling

Pedro C. S. Costa<sup>1</sup>✉, Philipp Schleich<sup>1,2,3</sup>, Mauro E. S. Morales<sup>4,5,6</sup> & Dominic W. Berry<sup>1</sup>

The solution of large systems of nonlinear differential equations is essential for many applications in science and engineering. We present three improvements to existing quantum algorithms based on the Carleman linearisation technique. First, we use a high-precision method for solving the linearised system that yields logarithmic dependence on the error and near-linear dependence on time. Second, we introduce a rescaling strategy that significantly reduces the cost, which would otherwise scale exponentially with the Carleman order, thus limiting quantum speedups for PDEs. Third, we derive tighter error bounds for Carleman linearisation. We apply our results to a class of discretised reaction-diffusion equations using higher-order finite differences for spatial resolution. We also show that enforcing a stability criterion independent of the discretisation can conflict with rescaling due to the mismatch between the max-norm and the 2-norm. Nonetheless, efficient quantum solutions remain possible when the number of discretisation points is constrained, as enabled by higher-order schemes.

Many processes in nature exhibit nonlinear behaviour that is not sufficiently approximated by linear dynamics. Examples range from biological systems, chemical reactions, fluid flow, and population dynamics to problems in climate science. Because the Schrödinger equation is linear, quantum algorithms are more naturally designed for linear ordinary differential equations (ODEs), as in refs. 1–9. These algorithms are normally based on discretising time to encode the ODE in a system of linear equations, then using quantum linear system solvers<sup>10,11</sup>. Others are based on a time-marching strategy, solving the ODE using a linear combination of unitary dynamics<sup>8,9</sup>. The advantage of these quantum algorithms is that they naturally provide an exponential speedup in the dimension (number of simultaneous equations), similar to the simulation of quantum systems, with the caveat that the solution is encoded in the amplitudes of a quantum state.

The most natural way to approximate quantum solutions of partial differential equations (PDEs) is to first discretise the PDE to construct an ODE, which can then be solved using a quantum ODE algorithm. Although one might expect an exponential speedup in the number of discretisation points (which would give the dimension for the ODE), this is not realised. This approach to solve PDEs typically has a more modest polynomial speedup over classical methods due to the norm or condition number of the

matrices resulting from the discretisation. Clader et al.<sup>12</sup> suggested using preconditioners, though later work found that the preconditioners did not significantly reduce the condition number. Childs et al.<sup>13</sup> approached this problem by using higher-order finite difference stencils as well as a pseudo-spectral method. Alternatively, one can use a wavelet-based preconditioner to achieve scaling independent of the condition number in some cases<sup>14</sup>. Jin et al.<sup>15,16</sup> introduce a new method using a variable transformation which provides solutions of PDEs in an equivalent frame using quantum simulation techniques.

Quantum algorithms for nonlinear differential equations were addressed in early work which had very large complexity<sup>17</sup>. Later proposals were based on the nonlinear Schrödinger equation<sup>18</sup>, or an exact mapping of the nonlinear Hamilton–Jacobi PDE into a linear PDE<sup>19,20</sup>. Possibly the most promising approach for the solution of nonlinear ODEs is based on Carleman linearisation<sup>21</sup>, which involves transforming the nonlinear differential equation into a linear differential equation on multiple copies of the vector. This approach can be realised particularly easily for differential equations with polynomial nonlinearities and has been applied to quantum algorithms in the case of a quadratic function as the nonlinear part of the ODEs<sup>22</sup>, for a higher power of the function for a specific PDE<sup>23</sup>, and for the notorious

<sup>1</sup>School of Mathematical and Physical Sciences, Macquarie University, Sydney, NSW, Australia. <sup>2</sup>Department of Computer Science, University of Toronto, Toronto, ON, Canada. <sup>3</sup>Vector Institute for Artificial Intelligence, Toronto, ON, Canada. <sup>4</sup>Centre for Quantum Software and Information, University of Technology Sydney, Sydney, NSW, Australia. <sup>5</sup>Centre for Quantum Computation and Communication Technology, University of Technology Sydney, Sydney, NSW, Australia. <sup>6</sup>Joint Center for Quantum Information and Computer Science (QulCS), University of Maryland, College Park, MD, USA. ✉e-mail: [pcs.costa@protonmail.com](mailto:pcs.costa@protonmail.com)

Navier-Stokes equations<sup>24</sup>. The homotopy perturbation method to tackle quadratic nonlinear equations in ref. 25 leads to similar equations as Carleman linearisation.

However, most approaches to quantum Carleman linearisation<sup>22,23</sup> applied to PDEs suffer from high error rates due to simple discretisation schemes for the underlying PDE in time and space. One work<sup>7</sup> does use an improved discretisation in time via a truncated Taylor series<sup>2</sup>. Using a finer discretisation to achieve a given accuracy results in higher complexity, typically due to the complexity depending on the matrix condition number. That can result in the complexity being the same or worse than that for classical solution. Another difficulty in the use of Carleman linearisation in prior work is that the component with the solution may have low probability to be measured. In this study, we provide three improvements over prior work. First, we use higher-order methods in the time evolution as well as for the spatial discretisation for PDEs. Second, we use rescaling in order to eliminate the problem of the low probability of the component with the solution for an intrinsic system of ODEs. (Krovi<sup>7</sup> mentioned a rescaling at one point, though their explanation is unclear and it is unclear if they are using it.) Third, we provide a tighter bound on the error in Carleman linearisation by explicitly bounding repeated integrals. In the case of a PDE, the appropriate stability condition is in terms of the max-norm. However, the interaction of the requirement of the rescaling with the Carleman linearisation and the stability requirement for the ODE solver means that a stronger stability criterion is needed to enable efficient solution.

It is important to note that in the case of PDEs the factor that is exponential in  $N$  in prior work<sup>23</sup> would give a large power in the number of grid points. Since a simple classical algorithm would have complexity linear in the number of grid points, the quantum speedup would be eliminated. Our work demonstrates that quantum computers can provide a sublinear complexity in the number of grid points for nonlinear PDEs, as well as establishing the limitations to this type of approach. We present an overview of the general solution procedure of nonlinear differential equations on quantum computers in relation to the present work in Fig. 1.

## Results

The main focus of this work is the treatment of nonlinear differential equations, when we have an arbitrary power  $M$  in the nonlinear ODE problem on quantum computers, that is

$$\frac{d\mathbf{u}}{dt} = F_1\mathbf{u} + F_M\mathbf{u}^{\otimes M}, \quad (1)$$

followed by its application to the nonlinear reaction-diffusion PDE,

$$\partial_t u(\mathbf{x}, t) = D\Delta u(\mathbf{x}, t) + cu(\mathbf{x}, t) + bu^M(\mathbf{x}, t). \quad (2)$$

We give a number of improvements to the solution of nonlinear ODEs and PDEs.

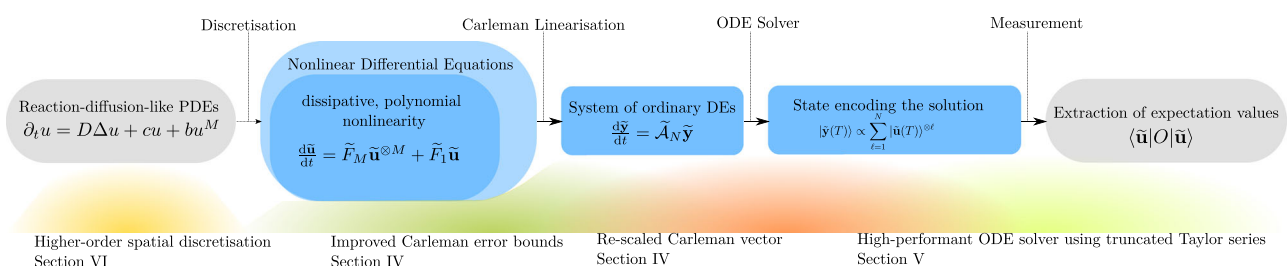
1. We use a higher-order method for discretisation of the PDE, which will be required in practice because the stability of the solution will require that the number of points is not too large.
2. We use rescaling of the components in the Carleman linearisation in order to ensure that the first component containing the solution can be obtained with high probability. We show that the amount of rescaling that can be used is closely related to the stability of the equations.
3. We provide a much tighter analysis of the error due to the Carleman linearisation for ODEs, and extend this analysis to PDEs. This analysis is dependent on the stability and the discretisation of the PDE.

All these improvements are dependent on the stability of the equations, which is required for the quantum algorithm to give an efficient solution. The equations have a linear dissipative term and the nonlinear growth term. As the input is made larger, the nonlinear term will cause growth and make the solution unstable. Therefore, for the solution to be dissipative, the input needs to be sufficiently small that the dissipative term dominates. In the ODE case the input is a vector  $\mathbf{u}_{\text{in}}$ , and the stability criterion can be given in terms of the 2-norm of that vector. In the case of the PDE, it is more appropriate to give the stability criterion in terms of the max-norm, because the 2-norm will change depending on the number of discretisation points.

Giving the stability criterion in terms of the max-norm then makes the analysis of higher-order discretisations challenging. The reason is that, while the first-order discretisation of the Laplace operator is stable in terms of the max-norm, the higher-order discretisations no longer are. In the analysis of the Carleman linearisation error it is required that the equations are stable. For the ODE this stability in terms of the 2-norm enables the 2-norm of the error to be bounded. For the PDE, stability in terms of the max-norm enables the max-norm to be bounded, but the higher-order discretisation complicates the analysis and means slightly stronger stability is required.

The reason why rescaling is needed is that the Carleman method involves constructing a quantum state with a superposition of one copy of the initial vector, two copies, and so forth up to  $N$  copies. If the initial vector is not normalised, then this means that there can be an exponentially large weight on the largest number of copies, whereas the first part of the superposition with a single copy is needed for the solution. In order to ensure the probability for obtaining that component is not exponentially small, the Carleman vector needs to be rescaled by (at least) the 2-norm so that there is sufficient weight on that first component. Even if  $N$  is small, this feature means that rescaling is essential in order to obtain any speedup over classical algorithms for PDEs. Without the rescaling, the complexity is superlinear in the number of grid points.

In order to ensure that the same equations are being solved, the components of the matrix need a matching rescaling, which can increase the weight of the nonlinear part (causing growth) as compared to the linear dissipative part. In the case of an ODE, we show that if the original nonlinear equation is dissipative then the linear ODE obtained from Carleman linearisation is also stable. That stability is required for the quantum ODE solver to be efficient. If the ODE is not stable, then the condition number will be exponentially large (in time), which causes the linear equation solver to have exponential complexity.



**Fig. 1 | Overview of solution pipeline for nonlinear (partial) differential equations in relation to presented contributions and location in paper.** The PDE is approximated by a vectorial ODE via discretisation,  $u \rightarrow \mathbf{u}$  and we use a scaling  $\mathbf{u} \rightarrow \tilde{\mathbf{u}}$  as described in Definition 1.

Similarly for the discretised PDE, there needs to be rescaling by the 2-norm in order to ensure there is adequate weight on the first component of the solution. The key difference now is that the stability of the equations is given in terms of the max-norm, but the rescaling is by the 2-norm which is typically larger. That rescaling can give a linear ODE that is no longer stable, which in turn would mean an exponential complexity of the algorithm. That is perhaps surprising, because the original nonlinear equation is stable.

However, if the PDE is sufficiently dissipative, then the discretised equation will still satisfy the stability criterion in terms of the 2-norm, and there will still be an efficient quantum algorithm. Because the 2-norm will increase without limit with the number of discretisation points, it is then crucial to minimise the number of discretisation points used. That further motivates using the higher-order discretisation of the PDE, because that minimises the number of discretisation points.

We now summarise the problem description and solution strategy follow by the main results for Eqs. (1) and (2).

### The ODE problem

Here, we present the problem of solving the nonlinear ODE, including the variable definitions and the dissipativity condition needed for an efficient quantum algorithm.

**Problem 1.** We consider the solution of a system of nonlinear (vectorial) dissipative ODEs of the form

$$\frac{d\mathbf{u}}{dt} = F_1\mathbf{u} + F_M\mathbf{u}^{\otimes M}, \quad (3)$$

with initial data

$$\mathbf{u}(t = 0) = \mathbf{u}_{\text{in}}, \quad (4)$$

where  $\mathbf{u} = (u_1 \cdots, u_n)^T \in \mathbb{R}^n$  with time-dependent components  $u_j = u_j(t)$  for  $t \in [0, T]$  and  $j \in [n]$ , using the notation  $[n] = \{1, 2, \dots, n\}$ . The matrices  $F_M \in \mathbb{R}^{n \times n^M}$ ,  $F_1 \in \mathbb{R}^{n \times n}$  are time-independent. We denote the eigenvalues of  $(F_1 + F_1^\dagger)/2$  by  $\lambda_j$  and the dissipativity condition means that  $\lambda_j < 0$ . Denoting the maximum eigenvalue by  $\lambda_0$ , we require that  $R < 1$ , where

$$R := \frac{\|F_M\| \cdot \|\mathbf{u}_{\text{in}}\|^{M-1}}{|\lambda_0|}. \quad (5)$$

The task is to output a state  $|\mathbf{u}\rangle$  encoding the solution to Eq. (3) at time  $T$ .

In the “Methods” section, we show that to solve Problem 1, we first map the finite-dimensional system of nonlinear differential equations in Eq. (3) to an infinite-dimensional, linear set of ODEs that can be truncated to some order  $N$ . This mapping is the Carleman linearisation technique<sup>21</sup>, which has previously been applied to quantum algorithms in refs. 7,22,23. Next, we show that by rescaling the linearised ODEs, we can reduce the complexity of the quantum algorithm. This is followed by improved error bounds due to Carleman linearisation for the rescaled variable and an estimate of the overall complexity for obtaining the solution of the truncated linearised ODE.

In contrast to refs. 7,22, we do not consider the driving term; on the other hand, we explore arbitrary nonlinear powers in the ODE problem rather than constrained to the quadratic case as in refs. 7,22. When we have an arbitrary power  $M$  in the nonlinear ODE it is more challenging to include the driving term  $F_0$ , because  $F_0$  will produce characteristics of a more general polynomial of order  $M$  as opposed to just a single component. Therefore, to analyse the driving term we would also need to consider a general polynomial of order  $M$  for the nonlinear part of the ODE problem. We leave that considerably more complicated analysis to future work.

The solution of a linearised form of Problem 1 relies on oracles for  $F_1$ ,  $F_M$ , and the initial vector. We show later in the “Methods” section, that the

complexity of the solution in terms of calls to oracles for  $F_1$  and  $F_M$  scales as

$$\mathcal{O}\left(\frac{1}{\sqrt{1 - R^{2/(M-1)}}} \frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|} \lambda_{F_1} TN \log\left(\frac{N}{\varepsilon}\right) \log\left(\frac{N\lambda_{F_1} T}{\varepsilon}\right)\right). \quad (6)$$

In this complexity,  $\varepsilon$  is the allowable error, and  $\lambda_{F_1}$  is the  $\lambda$ -value for block encoding  $F_1$  (with an extra assumption on the efficiency of the block encoding of  $F_M$ ). An important quantity here is the Carleman order  $N$ , which can be chosen logarithmically in the allowable error provided  $R < 1$ . For the complexity in terms of calls to the preparation of the initial vector, there is an extra factor of  $N$ , but the final log factor can be omitted, so the overall complexity is similar. Without the rescaling, there would be an extra factor in the complexity  $\mathcal{O}(\|\mathbf{u}_{\text{in}}\|^N)$  that is exponential in  $N$ . Even though  $N$  can be chosen logarithmic in the other parameters, that would still result in large complexity.

The result as given in ref. 23 has that problem. The complexity from ref. 23 is (using Eq. (4.2) of that work and replacing  $a$  in their notation with  $c$  in our notation)

$$\mathcal{O}\left(\frac{1}{G^2\varepsilon} s T^2 D^2 d^2 n^{4/d} N^3 \|\mathbf{u}_{\text{in}}\|^{2N} \text{poly}\left(\log\left(\frac{c D d M n^{1/d} N s T}{G\varepsilon}\right)\right)\right), \quad (7)$$

where  $G$  denotes the average  $\ell_2$  solution norm of the history state, and  $s$  is the maximum sparsity of  $F_1$ ,  $F_M$ . The factor  $\|\mathbf{u}_{\text{in}}\|^{2N}$  exponential in  $N$  is due to the higher-order components of the Carleman vector without rescaling. They also have a factor of  $T^2$  rather than  $T$ , which is due to using a simple forward Euler scheme in time. We also give a further improvement in the polynomial factor of  $N$ , with our scaling being  $N$  in comparison to their  $N^3$ .

### Carleman solver for the reaction-diffusion equation

A large system of ODEs of the form in Eq. (3) may arise from discretisation of partial differential equations. Specifically, we can derive the nonlinear differential equation resulting from the discretisation of a nonlinear reaction-diffusion PDE similar to ref. 23,

$$\partial_t u(\mathbf{x}, t) = D\Delta u(\mathbf{x}, t) + cu(\mathbf{x}, t) + bu^M(\mathbf{x}, t). \quad (8)$$

This equation will be stable according to a criterion that depends on the max-norm of  $u(\mathbf{x}, t)$ , in contrast to the condition for the ODE that is based on the 2-norm. Discretising this PDE into an ODE, the stability condition  $R < 1$  would be stronger and depend on the number of discretisation points. That condition is stronger than necessary for the PDE, but after we use Carleman linearisation to give a linear ODE it requires  $R < 1$  for stability. This means that the stability condition needed for the quantum algorithm is stronger than that for the original PDE.

We explore techniques of finite-difference methods with higher-order approximations for the spatial discretisation of the PDEs. Our improved nonlinear ODE solver is then applied to the reaction-diffusion equation Eq. (8), with  $F_1$  resulting from the Laplacian discretisation and  $F_M$  giving the non-linearity from the PDE. The overall procedure is illustrated in Fig. 1.

We then show in the “Methods” section, Corollary 6, that for this PDE, the overall cost for the solution in terms of calls to the oracles that block encode  $F_1$  and  $F_M$  is

$$\mathcal{O}\left(\frac{1}{\sqrt{1 - R^{2/(M-1)}}} \frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|} (d n^{2/d} + |c|) TN \log\left(\frac{N}{\varepsilon}\right) \log\left(\frac{N(d n^{2/d} + |c|) T}{\varepsilon}\right)\right), \quad (9)$$

where we have used  $n$  gridpoints in total for the spatial discretisation of the  $d$ -dimensional PDE given in Eq. (8).

Classically, it is less useful to perform linearisation by the Carleman procedure, because the system size grows exponentially with the truncation number  $N$  making the simulation prohibitively costly. In general, explicit time-stepping methods like forward Euler or Runge–Kutta schemes do not

**Table 1 | Summary of variable names and conventions used throughout the manuscript**

Symbol	Description
$\mathbf{x}, t, T$	Spatial coordinate, time, and final time
$\mathbf{u}$	Solution vector for the nonlinear ODE
$\mathbf{y}$	Carleman state vector: $\mathbf{y} = [\mathbf{u}, \mathbf{u}^{\otimes 2}, \dots, \mathbf{u}^{\otimes N}]$
$F_1, F_M$	Linear and nonlinear terms of the ODE
$N$	Carleman truncation order
$\mathcal{A}, \mathcal{A}_N$	Full and truncated Carleman system matrix
$A_j^{(i)}$	$(j, i)$ -block component of Carleman matrix
$\varepsilon$	Solution error
$\eta$	Error in Carleman vector (also due to linearisation)
$\Delta$	Laplace operator
$L, L_k$	Finite-difference Laplacians of stencil order $k$
$D, c, b$	PDE coefficients: diffusion, decay, and nonlinearity
$n, d$	Number of gridpoints and dimension of the PDE
$\mathbb{I}$	Identity matrix
$\lambda_j(\cdot)$	$j$ th eigenvalue of matrix argument
$\lambda_{(i)}$	Subnormalization constant for block encoding
$\gamma$	Rescaling factor
$\tilde{(\cdot)}$	Rescaled quantity
$R$	Nonlinearity-to-decay strength ratio
$f_{j,k,M}$	Bound function for Carleman truncation error
$\Omega_k$	Interval defining Carleman error regime
$s(\cdot)$	Sparsity of matrix or operator
$[N]$	Index set $\{1, 2, \dots, N\}$

Scalars are denoted by lowercase letters (e.g.,  $c$ ), vectors by bold lowercase letters (e.g.,  $\mathbf{u}$ ), and matrices or operators by uppercase letters (e.g.,  $A$ ). Norms refer to the 2-norm unless otherwise specified.

rely on linearisation of the underlying differential equations. However, (semi-)implicit schemes which exhibit more favourable numerical stability rely on inversion of the system. This either requires linearisation (e.g., Carleman or Koopman–von-Neumann schemes) or methods to solve nonlinear systems, such as Newton–Raphson, which rely on a good initial guess and require inversion of a Jacobian matrix.

## Discussion

In this study, we proposed a set of improvements to quantum algorithms for nonlinear differential equations via Carleman linearisation, eliminating some of the exponential scalings seen in prior work. We have examined both ODEs, and a class of nonlinear PDEs corresponding to reaction-diffusion equations<sup>23</sup>. These improvements include

- rescaling the original dynamics,
- a truncated Taylor series for the time evolution,
- higher-order spatial discretisation of the PDEs, and
- tighter bounds on the error of Carleman linearisation.

The rescaling boosts the success probability, needing exponentially fewer steps for the amplitude amplification to obtain the solution component of interest in the linearised ODE system. That is vital to enable the complexity of the PDE solver to be sublinear in the number of discretisation points. The solution approximation via the truncated Taylor method gives a near-linear dependence on  $T$ , the total evolution time. The higher-order spatial discretisation greatly improves the complexity of the quantum solution of PDEs, because it reduces the number of discretisation points needed, which is needed to avoid stability problems.

We show that the stability criterion for PDEs, rescaling, Carleman linearisation, and stability criterion for ODE solvers all interact in a way that makes the solution of PDEs more challenging than was appreciated in prior work. In particular, the stability criterion for PDEs is in terms of a max-norm, but rescaling by the 2-norm is required to obtain a reasonable probability for the correct component of the Carleman vector. But, rescaling by the 2-norm can make the resulting system of linearised equations unstable, which causes the ODE solver to have exponential complexity. If the discretised PDE is still stable in terms of the 2-norm, then the resulting quantum algorithm will still be efficient. Because the 2-norm increases as  $\sqrt{n}$  in the number of discretisation points, the number of those points should be made as small as possible, which is why it is crucial to use the higher-order discretisation.

In future work, one could devise a less restricted quantum algorithm for solving nonlinear PDEs via some other approach. The feature that the linearised equations can be unstable even though the nonlinear equation is stable suggests that an alternative linear equation solver may be efficient. The reason why the condition number is large (causing the inefficiency) is that the solution can grow exponentially over time, but for an initial vector that is not of the Carleman form. A solver that is able to take advantage of the restricted form of states could potentially be efficient.

Furthermore, there are a number of important generalisations that can be made to the type of differential equations. Instead of just including a nonlinear term of order  $M$ , one could include all nonlinear orders up to  $M$ . That could also be used to analyse the effect of driving because the method used for quadratic nonlinearities would produce nonlinearities at a range of orders. A further generalisation that could be considered is time-dependent differential equations. These generalisations can be made in a simple way in the quantum algorithm, but the analysis to bound the error would be considerably more complicated.

## Methods

We begin by summarising the notation and key variables used throughout the manuscript. Table 1 defines the principal symbols, their roles, and the conventions adopted. This summary is intended to assist the reader in navigating the derivations and algorithmic steps that follow.

### Quantum Carleman solver with rescaling and improved error bounds on Carleman truncation

**Background on Carleman linearisation.** We start with the Carleman linearisation for the initial value problem described by the  $n$ -dimensional equation with a nonlinearity of order  $M$  as given in Eq. (3). We recall the dissipativity assumption on  $F_1$ , i.e., the eigenvalues of  $(F_1 + F_1^\dagger)/2$  are purely negative. The quantity  $\lambda_0$ , the eigenvalue closest to zero, thus gives the weakest amount of dissipation. This way,  $R$  in Eq. (5) can be used to quantify the strength of the nonlinearity of the problem. As shown in ref. 22, there exists a quantum algorithm that can solve Eq. (3) efficiently whenever  $R < 1$ . Furthermore, for  $R \geq \sqrt{2}$ , the problem was shown to be intractable on quantum computers.

Next, we briefly outline the key idea of the Carleman linearisation. First, notice that

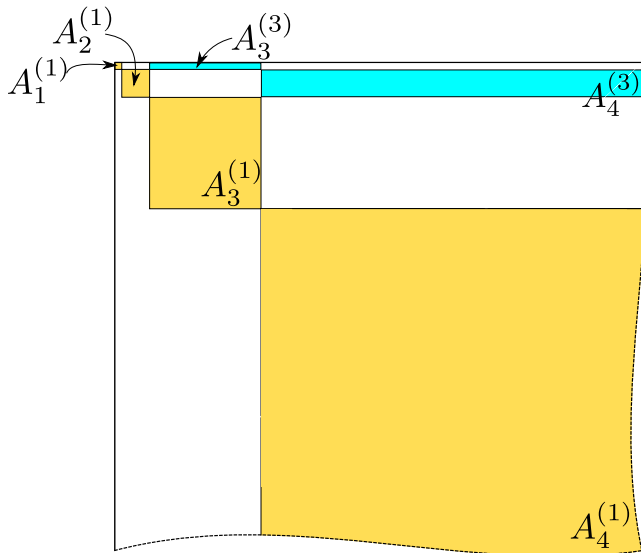
$$\mathbf{u}^{\otimes M} = (u_1^M, u_1^{M-1}u_2, \dots, u_1u_n^{M-1}, u_2u_1^{M-1}, \dots, u_n^{M-1}u_{n-1}, u_n^M)^T \in \mathbb{R}^{n^M}. \quad (10)$$

In particular, for  $M = 2$ , the Kronecker product gives

$$\mathbf{u}^{\otimes 2} = (u_1^2, u_1u_2, \dots, u_1u_n, u_2u_1, \dots, u_nu_{n-1}, u_n^2)^T \in \mathbb{R}^{n^2}. \quad (11)$$

Now, define a new variable consisting of Kronecker powers of the solution vector

$$\mathbf{y}_1 = \mathbf{u}, \mathbf{y}_2 = \mathbf{u}^{\otimes 2}, \dots, \mathbf{y}_N = \mathbf{u}^{\otimes N}, \dots, \quad (12)$$



**Fig. 2** | Depiction of a snippet of  $\mathcal{A}_N$  for  $M = 3$  until  $N = 4$ . Given the exponential increase in size, only a fraction of  $N = 4$  is shown. The diagonal blocks correspond to linear terms of the ODE, the upper-diagonal blocks to a nonlinearity on the  $(M - 1)$  st off-diagonal.

which we can summarise as a vector  $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N, \dots]^T$ . If we consider the time-derivative, we can identify the time-independent matrices  $F_M \in \mathbb{R}^{n \times n^M}$  and  $F_1 \in \mathbb{R}^{n \times n}$  as follows,

$$\begin{aligned} \frac{d\mathbf{y}_j}{dt} &= \frac{d\mathbf{u}^{\otimes j}}{dt} \\ &= \frac{d\mathbf{u}}{dt} \otimes \mathbf{u} \otimes \dots \otimes \mathbf{u} + \dots + \mathbf{u} \otimes \mathbf{u} \otimes \dots \otimes \frac{d\mathbf{u}}{dt} \\ &= (F_M \mathbf{u}^{\otimes M}) \otimes \mathbf{u} \otimes \dots \otimes \mathbf{u} + \dots + \mathbf{u} \otimes \mathbf{u} \otimes \dots \otimes (F_M \mathbf{u}^{\otimes M}) \\ &\quad + (F_1 \mathbf{u}) \otimes \mathbf{u} \otimes \dots \otimes \mathbf{u} + \dots + \mathbf{u} \otimes \mathbf{u} \otimes \dots \otimes (F_1 \mathbf{u}). \end{aligned} \quad (13)$$

We can write this in compact form,

$$\frac{d\mathbf{y}_j}{dt} = A_{j+M-1}^{(M)} \mathbf{y}_{j+M-1} + A_j^{(1)} \mathbf{y}_j, \quad (14)$$

where  $A_{j+M-1}^{(M)} \in \mathbb{R}^{n^j \times n^{j+M-1}}$  and  $A_j^{(1)} \in \mathbb{R}^{n^j \times n^j}$  with

$$\begin{aligned} A_{j+M-1}^{(M)} &= F_M \otimes \mathbb{I}^{\otimes(j-1)} + \mathbb{I} \otimes F_M \otimes \mathbb{I}^{\otimes(j-2)} + \dots + \mathbb{I}^{\otimes(j-1)} \otimes F_M \\ A_j^{(1)} &= F_1 \otimes \mathbb{I}^{\otimes(j-1)} + \mathbb{I} \otimes F_1 \otimes \mathbb{I}^{\otimes(j-2)} + \dots + \mathbb{I}^{\otimes(j-1)} \otimes F_1, \end{aligned} \quad (15)$$

where the  $\mathbb{I}$  operation is the identity with the same domain as  $F_1$ , i.e.,  $\mathbb{R}^{n \times n}$ .

This results in an infinite-dimensional linear system, as there is no bound on the range of  $j$ . To make this computationally feasible, we restrict to  $j \in [N]$  for some  $N \ni N > M$ . Further, we can see that  $N > M$  is a requirement in order to be able to capture any effects coming from a nonlinearity of order  $M$ . This allows one to write down a matrix form,

$$\frac{d\mathbf{y}}{dt} = \mathcal{A}_N \mathbf{y}, \quad (16)$$

with

$$\mathcal{A}_N = \begin{bmatrix} A_1^{(1)} & 0 & \dots & 0 & A_M^{(M)} & 0 & \dots & 0 \\ 0 & A_2^{(1)} & \dots & 0 & 0 & A_{M+1}^{(M)} & 0 & \vdots \\ \vdots & 0 & \ddots & & & 0 & \ddots & 0 \\ & & \ddots & \ddots & & & \ddots & A_N^{(M)} \\ & & & \ddots & \ddots & & & 0 \\ & & & & \ddots & \ddots & & \vdots \\ \vdots & & & & & 0 & A_{N-1}^{(1)} & 0 \\ 0 & 0 & \dots & & \dots & 0 & A_N^{(1)} \end{bmatrix}. \quad (17)$$

The matrix  $\mathcal{A}_N \in \mathbb{R}^{N_{\text{tot}} \times N_{\text{tot}}}$  is called the Carleman matrix with truncation order  $N$ , where  $N_{\text{tot}} = \sum_{j=1}^N n^j = \frac{n(n^N - 1)}{n - 1}$ . The non-truncated, infinitely large matrix we call  $\mathcal{A}$ . As the dimensionality of the system is exponential in the order of Carleman truncation (see Fig. 2), this technique tends to be intractable for practical applications on classical computers.

The simple block structure of the matrix  $\mathcal{A}_N$  enables us to obtain the upper bound for  $\|\mathcal{A}_N\|$  in terms of the norms of the submatrix of  $\mathcal{A}_N$ , that is

$$\begin{aligned} \|\mathcal{A}_N\| &\leq \max_{1 \leq j \leq N} \|A_j^{(1)}\| + \max_{1 \leq j \leq N+M-1} \|A_j^{(M)}\| \\ &= N\|F_1\| + (N - M + 1)\|F_M\|. \end{aligned} \quad (18)$$

A similar relation holds for the  $\lambda$ -values, which is important for the estimation of the complexity of our quantum algorithm. In what follows, we present a lemma that allows us to quantify the total error involved in the Carleman truncation. Our lemma considers the error from the Carleman linearisation for the rescaled nonlinear ODE problem when we have an arbitrary power  $M$  for the function, as opposed to the quadratic case without the rescaling given in ref. 22. To that end, we will first present said rescaling.

**A rescaled Carleman solver.** We will motivate this rescaling by looking at the measurement probabilities of components in the vector  $\mathbf{y} = [\mathbf{u}, \mathbf{u}^{\otimes 2}, \dots, \mathbf{u}^{\otimes N}]^T$ . Recall that the sole entry we are interested in measuring will be  $\mathbf{y}_1 \equiv \mathbf{u}$ . The standard way to encode the solution  $\mathbf{u}(t)$  in a computational basis  $\{|j\rangle\}$  is

$$|\mathbf{u}(t)\rangle = \sum_{j=1}^n u_j(t) |j\rangle. \quad (19)$$

Analogously, components  $|\mathbf{y}_m\rangle$  of  $\mathbf{y}$  are written as a quantum state as

$$\begin{aligned} |\mathbf{y}_m(t)\rangle &= \sum_{j_1, \dots, j_m=1}^n u_{j_1}(t) \dots u_{j_m}(t) |m, j_1 \dots j_m, 0^{\otimes \log(n)(N-m)}\rangle \\ &= \sum_{j_1, \dots, j_m=1}^n u_{j_1}(t) \dots u_{j_m}(t) |y_m^{(j_1 j_2 \dots j_m)}\rangle, \end{aligned} \quad (20)$$

with

$$|y_m^{(j_1 j_2 \dots j_m)}\rangle := |m, j_1 \dots j_m, 0^{\otimes \log(n)(N-m)}\rangle. \quad (21)$$

This follows the state encoding outlined in the Supplementary Information 3.C in ref. 22, where in each step up to the largest order  $N$ , extra dimensions are padded in the form of  $|0\rangle$ 's to avoid the structure of a superposition over components of different size. The first register is set to  $m$  so we can distinguish the order by measurement of a subsystem. Then, we can write the



full vector  $|\mathbf{y}(t)\rangle$  as follows:

$$|\mathbf{y}(t)\rangle = \sum_{j_1=1}^n u_{j_1}(t) |y_1^{(j_1)}\rangle + \sum_{j_1, j_2=1}^n u_{j_1}(t) u_{j_2}(t) |y_2^{(j_1 j_2)}\rangle + \cdots + \sum_{j_1, \dots, j_N=1}^n u_{j_1}(t) \dots u_{j_N}(t) |y_N^{(j_1 j_2 \dots j_N)}\rangle. \quad (22)$$

For a normalised quantum state, the amplitudes  $u_{j_i}(t)$  in Eq. (22) need to be normalised so that  $\langle \mathbf{y} | \mathbf{y} \rangle = 1$ . We then have to consider the normalisation factor  $1/\sqrt{V_N}$  where

$$V_N = \|\mathbf{y}\|^2 = \sum_{\ell=1}^N \|\mathbf{u}(t)\|^{2\ell} = \|\mathbf{u}(t)\|^2 \frac{1 - \|\mathbf{u}(t)\|^{2N}}{1 - \|\mathbf{u}(t)\|^2}. \quad (23)$$

Note that this formula does not work in the case that  $\|\mathbf{u}(t)\| = 1$ . We therefore adopt the convention that wherever there appears a ratio of this form, for  $\|\mathbf{u}(t)\| = 1$  it takes the value in the limit  $\|\mathbf{u}(t)\| \rightarrow 1$ , so

$$V_N \rightarrow N. \quad (24)$$

The solution of the nonlinear ODE is given by the first component, where the probability is given by

$$P(\mathbf{y}_1(t)) = \sum_{j_1=1}^n \left| \langle y_1^{(j_1)} | \mathbf{y}(t) \rangle \right|^2 = \frac{1}{V_N} \sum_{j_1=1}^n |u_{j_1}(t)|^2 = \frac{1 - \|\mathbf{u}(t)\|^2}{1 - \|\mathbf{u}(t)\|^{2N}}. \quad (25)$$

From this equation, we see that as we increase the Carleman truncation order we also increase  $V_N$ , which suppresses the probability of extracting the desired component. This brings an exponential cost in  $N$  for the algorithm due to the  $\mathcal{O}(1/\sqrt{P(\mathbf{y}_1(t))})$  rounds of amplitude amplification needed at the end. To avoid this high cost in the algorithm, we propose the following rescaling, which can significantly reduce the cost of amplitude amplification.

**Definition 1. (Rescaled Carleman problem).** Consider a nonlinear ODE system of the form  $\frac{d\mathbf{u}}{dt} = F_1 \mathbf{u} + F_M \mathbf{u}^{\otimes M}$  as in Problem 1. Then, using a variable transformation in the form of a rescaling  $\tilde{\mathbf{u}} = \mathbf{u}/\gamma$  with  $\gamma > 0$ , we obtain another system in the rescaled variable

$$\frac{d\tilde{\mathbf{u}}}{dt} = \tilde{F}_1 \tilde{\mathbf{u}} + \tilde{F}_M \tilde{\mathbf{u}}^{\otimes M}, \quad (26)$$

with  $\tilde{F}_1 = F_1$  and  $\tilde{F}_M = \gamma^{M-1} F_M$ .

This allows us to improve the measurement probability in the following sense.

**Lemma 2. (Measurement probability of the rescaled Carleman problem).** Using the rescaling in Definition 1, using a scaling factor  $\gamma \geq \|\mathbf{u}_{\text{in}}\|$  and assuming dissipativity of the ODE, the probability to measure  $\tilde{\mathbf{u}} = \tilde{\mathbf{y}}_1$  is given by

$$P(\tilde{\mathbf{y}}_1(t)) = \frac{1 - \frac{\|\mathbf{u}(t)\|^2}{\gamma^2}}{1 - \left(\frac{\|\mathbf{u}(t)\|}{\gamma}\right)^{2N}} \geq \frac{1}{N}. \quad (27)$$

**Proof.** Using the rescaling  $\gamma > 0$ , we obtain a new normalisation

$$\tilde{V}_N = \sum_{i=1}^N \left( \frac{\|\mathbf{u}(t)\|}{\gamma} \right)^{2i} = \|\tilde{\mathbf{u}}(t)\|^2 \frac{1 - \|\tilde{\mathbf{u}}(t)\|^{2N}}{1 - \|\tilde{\mathbf{u}}(t)\|^2}, \quad (28)$$

with  $\|\tilde{\mathbf{u}}(t)\| = \|\mathbf{u}(t)\|/\gamma$ . Given dissipativity of the ODE, we have  $\|\mathbf{u}(t)\| \leq \|\mathbf{u}_{\text{in}}\|$ , so  $\|\tilde{\mathbf{u}}(t)\| \leq 1$ . In turn that implies

$$\frac{1 - \|\tilde{\mathbf{u}}(t)\|^{2N}}{1 - \|\tilde{\mathbf{u}}(t)\|^2} \leq N. \quad (29)$$

The measurement probability to obtain  $\tilde{\mathbf{y}}_1(t)$  is then

$$P(\tilde{\mathbf{y}}_1(t)) = \sum_{j_1=1}^n \left| \langle y_1^{(j_1)} | \tilde{\mathbf{y}}(t) \rangle \right|^2 = \frac{\|\tilde{\mathbf{u}}(t)\|^2}{\tilde{V}_N} = \frac{1 - \|\tilde{\mathbf{u}}(t)\|^2}{1 - \|\tilde{\mathbf{u}}(t)\|^{2N}} \geq \frac{1}{N}. \quad (30)$$

Therefore, using the parameter  $\gamma$ , we can adjust the probability to obtain  $\tilde{\mathbf{y}}_1$ . Here we have taken  $\gamma \geq \|\mathbf{u}_{\text{in}}\|$ , though the first expression does not depend on this assumption. The probability is equal to  $1/N$  if  $\gamma = \|\mathbf{u}_{\text{in}}\| = \|\mathbf{u}(t)\|$ , and otherwise for  $\gamma > \|\mathbf{u}(t)\|$  the probability is even better. Thus the rescaling avoids the exponential (in  $N$ ) suppression of the probability of obtaining the component of interest of the ODE problem, which occurs for  $\|\mathbf{u}(t)\| > 1$  without rescaling.

When we apply the rescaling above into Eq. (15) we obtain a linearised system in the rescaled solution vector with  $\tilde{A}_j^{(1)} = A_j^{(1)}$  and  $\tilde{A}_{j+M-1}^{(M)} = \gamma^{M-1} A_{j+M-1}^{(M)}$ , and as a result we can write the rescaled Carleman linearisation as

$$\frac{d\tilde{\mathbf{y}}}{dt} = \tilde{\mathcal{A}}_N \tilde{\mathbf{y}}, \quad (31)$$

where

$$\tilde{\mathcal{A}}_N = \begin{bmatrix} A_1^{(1)} & 0 & \cdots & 0 & \gamma^{M-1} A_M^{(M)} & 0 & \cdots & 0 \\ 0 & A_2^{(1)} & \cdots & 0 & 0 & \gamma^{M-1} A_{M+1}^{(M)} & 0 & \vdots \\ \vdots & 0 & \ddots & & & 0 & \ddots & 0 \\ & & \ddots & \ddots & & & \ddots & \gamma^{M-1} A_N^{(M)} \\ & & & \ddots & & & & 0 \\ & & & & \ddots & & & \vdots \\ \vdots & & & & & 0 & A_{N-1}^{(1)} & 0 \\ 0 & 0 & \cdots & & & \cdots & 0 & A_N^{(1)} \end{bmatrix}. \quad (32)$$

We discuss the cost of implementing the rescaled dynamics when introducing the ODE solver.

**Error bounds on rescaled solution.** Next, we present error bounds on the global and component-wise errors due to Carleman linearisation in Lemma 3 and Lemma 4, where we make use of the rescaling technique outlined in the previous section. The first lemma provides a bound on the overall error in the Carleman vector. The error bounds we present here are based on the 2-norm.

**Lemma 3. (Global rescaled Carleman error).** Consider the ODE from Eq. (3) with its Carleman linearisation in Eq. (13) truncated at order  $N$ . Let  $F_1$  be dissipative, so that for  $\lambda_0 < 0$  with  $|\lambda_0| > \|\mathbf{u}_{\text{in}}\|^{M-1} \|F_M\|$  and therefore  $\|\tilde{\mathbf{u}}_{\text{in}}\| \geq \|\tilde{\mathbf{u}}(t)\|$  for  $t > 0$ . Then, the error in the rescaled solution as defined in Lemma 2 is given by  $\eta_j = \tilde{\mathbf{u}}^{\otimes j} - \tilde{\mathbf{y}}_j$  at order  $j \in [N]$  due to Carleman truncation  $N > M \geq 2$  and a scaling factor  $\gamma = \|\mathbf{u}_{\text{in}}\|$ ;  $\tilde{\mathbf{u}}$  denotes the exact solution to the underlying ODE whereas  $\tilde{\mathbf{y}}$  is the approximation due to Carleman truncation. Then, this error for any  $j \in [N]$  is upper bounded by the overall error vector,

$$\|\eta_j(t)\| \leq \|\eta(t)\| \leq (M-1) \|F_M\| \|\mathbf{u}_{\text{in}}\|^{M-1} \frac{1 - e^{N(\lambda_0 + \gamma^{M-1} \|F_M\|)t}}{|\lambda_0 + \gamma^{M-1} \|F_M\||}. \quad (33)$$

The detailed proof is presented in Supplementary Section II of the Supplementary Information. Related results were given in ref. 22 and ref. 7. Neither included a general power for the nonlinearity, and were restricted to  $M = 2$ . Furthermore, we provide an exponential reduction in the Carleman order dependence due to the rescaling, i.e.,  $\|\eta\| \propto \|\mathbf{u}_{\text{in}}\|^M$  in opposed to  $\|\mathbf{u}_{\text{in}}\|^N$ . Although ref. 7 mentioned rescaling, it appears not to have been used in the error analysis. If the rescaled form was being used in that work, then it would imply that  $\|\mathbf{u}_{\text{in}}\|$  would be equal to 1, so  $\log(1/\|\mathbf{u}_{\text{in}}\|) = 0$  which results in  $N$  being infinite in Eq. (7.23) of ref. 7.

A problem with using this form is that it does not go down with the Carleman order. We aim to show that the error may be made arbitrarily small with higher-order Carleman approximations. We can provide tighter bounds when we consider the individual components of the Carleman vector, as in the following lemma.

**Lemma 4. (Component-wise Carleman error).** Under the same setting as in Lemma 3, and  $j \in [N]$ , the Carleman error for each individual component of  $\eta_j$  satisfies

$$\|\eta_j(t)\| \leq \left(\frac{\|\mathbf{u}_{\text{in}}\|}{\gamma}\right)^j R^k f_{j,k,M}(|\lambda_0|t), \quad j \in \Omega_k \quad (34)$$

where

$$f_{j,k,M}(\tau) = 1 - \frac{(M-1)\Gamma(k+j/(M-1))}{(k-1)!\Gamma(j/(M-1))} \sum_{\ell=0}^{k-1} (-1)^\ell \binom{k-1}{\ell} \frac{e^{-(\ell(M-\ell+j)\tau)}}{\ell M - \ell + j}, \quad (35)$$

for  $k \in \{1, 2, \dots, \lceil \frac{N}{M-1} \rceil\}$  and  $k$  is determined so that for any  $j$ , we have  $k$  whenever  $j$  falls into the index set  $j \in \Omega_k$  with

$$\Omega_k := \{N - k(M-1) + 1, \dots, N + (k-1)(1-M)\}. \quad (36)$$

In particular, for  $k = \lceil N/(M-1) \rceil$  we have

$$\|\eta_1(t)\| \leq \frac{\|\mathbf{u}_{\text{in}}\|}{\gamma} R^{\lceil \frac{N}{M-1} \rceil} f_{1, \lceil N/(M-1) \rceil, M}(|\lambda_0|t). \quad (37)$$

The proof of Lemma 4 can be found in Supplementary Section II B of the Supplementary Information. The function  $f_{j,k,M}(\tau)$  is monotonically decreasing with  $k$ , and in particular  $f_{j,k,M}(\tau) \leq f_{j,1,M}(\tau) = 1 - e^{-j\tau}$  (see Supplementary Section II B of the Supplementary Information). This result does not depend on the choice of rescaling  $\gamma$ . There is a factor of  $1/\gamma^j$  in the definition of  $\eta_j$ , so the result is effectively independent of the choice of rescaling. Moreover,  $\|\eta_1(t)\|$  gives the error in the desired component at the end, and shows that the error in  $\mathbf{u}$  is proportional to  $\|\mathbf{u}_{\text{in}}\|$ .

A similar result was provided in ref. 23 without using the rescaling, though that does not affect the result for the error. We give a significant improvement over the result in ref. 23 by evaluating the nested integrals to give the function  $f_{j,k,M}(\tau)$ , whereas the result in ref. 23 just corresponds to replacing  $f_{j,k,M}(\tau)$  with its upper bound of 1.

We can use Lemma 4 to solve for a lower bound on  $N$  for a given allowable error. In practice, we are interested in the error in the solution relative to  $\|\mathbf{u}_{\text{in}}\|$  rather than  $\gamma$ , so we aim to bound  $\|\eta_1(t)\|/\|\mathbf{u}_{\text{in}}\|$ . Given a maximum allowable error  $\varepsilon$ , we then require

$$\varepsilon \leq R^{\lceil \frac{N}{M-1} \rceil} f_{1, \lceil N/(M-1) \rceil, M}(|\lambda_0|t) \leq R^{\lceil \frac{N}{M-1} \rceil}. \quad (38)$$

It is therefore sufficient to choose  $N$  as

$$\left\lceil \frac{N}{M-1} \right\rceil \leq \frac{\log(1/\varepsilon)}{\log(1/R)}, \quad (39)$$

or

$$N = (M-1) \left\lceil \frac{\log(1/\varepsilon)}{\log(1/R)} \right\rceil - (M-2). \quad (40)$$

We can also numerically solve for  $N$ , by using the exact expression for  $f_{j,k,M}(\tau)$  given in Eq. (35). That will give a tighter lower bound on  $N$ , but there is not a closed-form expression.

### Solution of the linearised system of ordinary differential equations using a truncated Taylor series

Next, we describe how to solve the system of ODEs that results from the Carleman mapping applied onto the nonlinear system. The most simple way to solve the system of ODEs is to apply the first-order method for time discretisation known as the explicit Euler method. Upon application of the Euler method, there is a linear system of equations that can be solved. Here, this is a quantum linear system problem (QLSP), as the solution is encoded in a quantum state. In what follows, we aim to solve the linear system by a more sophisticated method than explicit Euler. The main drawback of the forward Euler method is low accuracy since it is a first-order method, meaning finer time discretisation is required to achieve a required precision. As a result, the dependence of the complexity for solving the QLSP is quadratic in the solution time, and there is a near-linear factor in the inverse error<sup>22,23</sup>.

Here, we follow the procedure outlined in ref. 3, which allows us to obtain an algorithm that has complexity near-linear in time and logarithmic in the inverse error. The solution of a time-independent ODE system

$$\frac{d\mathbf{u}(t)}{dt} = A\mathbf{u}(t), \quad (41)$$

may be approximated by  $\mathbf{u}_K(t) = W_K(t, t_0)\mathbf{u}(t_0)$ , with

$$W_K(t, t_0) := \sum_{\ell=0}^K \frac{(A\Delta t)^\ell}{\ell!}. \quad (42)$$

This is a Taylor series truncated at order  $K$ . The error in the solution due to time propagation can be bounded as

$$\|\mathbf{u}_K(t) - \mathbf{u}(t)\| \in \mathcal{O}\left(\frac{(\|A\|\Delta t)^{K+1}}{(K+1)!} \|\mathbf{u}(t_0)\|\right). \quad (43)$$

We aim to solve Eq. (16) where the vector  $\mathbf{u}(t)$  is mapped to a rescaled vector  $\tilde{\mathbf{y}}(t)$  and  $A$  is the rescaled Carleman matrix  $U_{\tilde{A}_N}$  truncated at order  $N$ .

Following Theorem 2 in ref. 3, there exists a quantum algorithm that can provide an approximation  $|\tilde{\mathbf{y}}\rangle$  of the solution  $|\tilde{\mathbf{y}}(T)\rangle$  satisfying  $\| |\tilde{\mathbf{y}}\rangle - |\tilde{\mathbf{y}}(T)\rangle \| \leq \varepsilon \gamma_{\text{max}}$ . To do so, we require that  $\mathcal{A}_N$  has non-positive logarithmic norm and we have the oracles  $U_y$  to prepare the initial state and block encoding of  $\mathcal{A}_N$  via  $U_{\mathcal{A}_N}^-$  with  $|0\rangle_{\mathcal{A}_N} \langle 0| = \tilde{\mathcal{A}}_N / \lambda_{\mathcal{A}_N}$ . Then, to achieve the desired accuracy, the average number of calls to  $U_y$  and  $U_{\mathcal{A}_N}^-$  needed are

$$U_y : \quad \mathcal{O}\left(\tilde{\mathcal{R}}_{\mathcal{A}_N} \lambda_{\mathcal{A}_N} T \log\left(\frac{1}{\varepsilon}\right)\right) \quad (44)$$

$$U_{\mathcal{A}_N}^- : \quad \mathcal{O}\left(\tilde{\mathcal{R}}_{\mathcal{A}_N} \lambda_{\mathcal{A}_N} T \log\left(\frac{1}{\varepsilon}\right) \log\left(\frac{\lambda_{\mathcal{A}_N}}{\varepsilon}\right)\right). \quad (45)$$

Furthermore, the number of additional elementary gates scales as

$$\mathcal{O}\left(\tilde{\mathcal{R}}_{\mathcal{A}_N} T \log\left(\frac{1}{\varepsilon}\right) \log^2\left(\frac{\tilde{\mathcal{A}}_N^T}{\varepsilon}\right)\right). \quad (46)$$

In these expressions

$$\tilde{\mathcal{R}} \geq \frac{y_{\max}}{\|\tilde{\mathbf{y}}(T)\|} \quad (47)$$

$$y_{\max} \geq \max_{t \in [0, T]} \|\tilde{\mathbf{y}}(t)\|. \quad (48)$$

The stability requirement on the ODE to use the solver as in ref. 3 is that the logarithmic norm of the matrix is non-positive (similar to ref. 7). That norm is given by the eigenvalues of  $(\tilde{\mathcal{A}}_N + \tilde{\mathcal{A}}_N^\dagger)/2$ . The eigenvalues of that matrix can be bounded via the block form of the Gershgorin circle theorem. That is equivalent to the usual Gershgorin circle theorem, except using the spectral norms of the off-diagonal blocks. For example, see Theorem 2 of ref. 26, or ref. 27.

For  $(\tilde{\mathcal{A}}_N + \tilde{\mathcal{A}}_N^\dagger)/2$  we obtain rows with  $A_j^{(1)}$  and  $\gamma^{M-1}A_j^{(M)}/2$  (for  $j \geq M$ ) and  $\gamma^{M-1}A_{j+M-1}^{(M)}/2$  (for  $j + M - 1 \leq N$ ). Now  $\|A_{j+M-1}^{(M)}\| \leq j \|F_M\|$ , so the sum of the norms of the off-diagonal blocks is at most, for  $j \geq M$  and  $j + M - 1 \leq N$ ,

$$\|A_{j+M-1}^{(M)}\| + \|A_j^{(M)}\| \leq j \|F_M\| + (j - M + 1) \|F_M\| = (2j - M + 1) \|F_M\|. \quad (49)$$

In the case  $j < M$  but  $j + M - 1 \leq N$  then we get  $j \|F_M\|$ . If  $j + M - 1 > N$  but  $j \geq M$  then we get  $(j - M + 1) \|F_M\|$ . Now the maximum eigenvalue of  $[A_j^{(1)} + (A_j^{(1)})^\dagger]/2$  is  $j\lambda_0$ . In that case the eigenvalues of  $(\tilde{\mathcal{A}}_N + \tilde{\mathcal{A}}_N^\dagger)/2$  can be at most

$$\begin{cases} j\lambda_0 + j\gamma^{M-1}\|F_M\|/2, & 0 < j < M \\ j\lambda_0 + (2j - M + 1)\gamma^{M-1}\|F_M\|/2, & j \geq M \text{ and } j \leq N - M + 1 \\ j\lambda_0 + (j - M + 1)\gamma^{M-1}\|F_M\|/2, & N \geq j > N - M + 1 \end{cases} \quad (50)$$

We can then see that the eigenvalues will be non-positive given all three inequalities

$$\gamma^{M-1} \leq \frac{2|\lambda_0|}{\|F_M\|}, \quad (51)$$

$$\gamma^{M-1} \leq \frac{|\lambda_0|}{[1 - (M - 1)/(2(N - M + 1))]\|F_M\|}, \quad (52)$$

$$\gamma^{M-1} \leq \frac{|\lambda_0|}{[1 - (M - 1)/N]\|F_M\|}. \quad (53)$$

Provided  $N \geq 2(M - 1)$  (as would normally be the case) the middle inequality would imply the other two. In all cases we can satisfy these inequalities using

$$\gamma^{M-1} \leq \frac{|\lambda_0|}{\|F_M\|} = \frac{\|\mathbf{u}_{\text{in}}\|^{M-1}}{R}, \quad (54)$$

or

$$\gamma \leq \frac{\|\mathbf{u}_{\text{in}}\|}{R^{1/(M-1)}}, \quad (55)$$

where we used the definition of  $R$  from Eq. (5) in the equality above. Berry and Costa<sup>3</sup> argue that for cases where the solution does not decay

significantly,  $\mathcal{R} \in \mathcal{O}(1)$ . Here, we consider dissipative dynamics without driving, so  $\mathcal{R}$  may be large. That is less of a problem for driven equations. We expect that our methods can be applied to driven equations as well, but the error analysis is considerably more complicated so we leave it as a problem for future work.

We can construct the block encoding of the Carleman matrix  $\tilde{\mathcal{A}}_N$  in terms of the block encoding of  $F_1$  and  $F_M$ , as discussed in Supplementary Section VII of the Supplementary Information. Denoting the values of  $\lambda$  for  $F_1$  and  $F_M$  by  $\lambda_{F_1}$  and  $\lambda_{F_M}$  respectively, the value of  $\lambda$  for  $\tilde{\mathcal{A}}_N$  is

$$\lambda_{\tilde{\mathcal{A}}_N} \leq N\lambda_{F_1} + (N - M + 1)\gamma^{M-1}\lambda_{F_M}. \quad (56)$$

This expression easily follows from expressing  $\tilde{\mathcal{A}}_N$  as a sum, and the value of  $\lambda$  being the sum of the values of  $\lambda$  in the sum. Since  $\tilde{\mathcal{A}}_N$  includes  $A_j^{(1)}$  up to  $A_N^{(1)}$ , and  $A_N^{(1)}$  is a sum of  $N$  operators with identity tensored with  $F_1$ , we obtain the term  $N\lambda_{F_1}$  above. Similarly, we have  $\gamma^{M-1}A_j^{(M)}$  up to  $\gamma^{M-1}A_N^{(M)}$ , and  $A_N^{(M)}$  is a sum of  $N - M + 1$  operators with  $F_M$ , giving the  $(N - M + 1)\gamma^{M-1}\lambda_{F_M}$  term.

If we choose  $\gamma^{M-1} = |\lambda_0|/\|F_M\|$  as above, then

$$N\|F_1\| > (N - M + 1)\gamma^{M-1}\|F_M\|. \quad (57)$$

In typical cases we would expect that  $\lambda_{F_1} \propto \|F_1\|$  and  $\lambda_{F_M} \propto \|F_M\|$ . That would imply

$$\lambda_{\tilde{\mathcal{A}}_N} \lesssim 2N\lambda_{F_1}. \quad (58)$$

Note that the scaling has not increased the value of  $\lambda$  by more than a constant factor. Note that this is assuming that the  $\lambda$ -values and norms in the block encoding are comparable, so it is possible it could be violated if the block encoding of  $F_M$  is inefficient, so  $\lambda_{F_M}$  is much larger than  $\|F_M\|$ .

Now for  $\mathcal{R}$  we have  $y_{\max}$  which considers the maximum norm that the vector can assume along the entire time evolution. Since we are working with a dissipative problem the maximum occurs at  $t = 0$ . First we consider the case without the scaling for comparison. To compute the norm  $\|\mathbf{y}(0)\|$ , note that it is the vector resulting from the Carleman mapping, i.e.,  $\mathbf{y}(0) = [\mathbf{u}_{\text{in}}, \mathbf{u}_{\text{in}}^{\otimes 2}, \dots, \mathbf{u}_{\text{in}}^{\otimes N}]^T$ , so

$$\|\mathbf{y}(0)\|^2 = \|\mathbf{u}_{\text{in}}\|^2 \frac{1 - \|\mathbf{u}_{\text{in}}\|^{2N}}{1 - \|\mathbf{u}_{\text{in}}\|^2}, \quad (59)$$

as in Eq. (23). Similarly for the value of the norm at time  $T$ ,

$$\|\mathbf{y}(T)\|^2 = \|\mathbf{u}(T)\|^2 \frac{1 - \|\mathbf{u}(T)\|^{2N}}{1 - \|\mathbf{u}(T)\|^2}. \quad (60)$$

Therefore

$$\begin{aligned} \mathcal{R} &\geq \frac{y_{\max}}{\|\mathbf{y}(T)\|} \\ &= \left[ \frac{(1 - \|\mathbf{u}_{\text{in}}\|^{2N})(1 - \|\mathbf{u}(T)\|^2)}{(1 - \|\mathbf{u}_{\text{in}}\|^2)(1 - \|\mathbf{u}(T)\|^{2N})} \right]^{1/2} \frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}. \end{aligned} \quad (61)$$

Moreover, the above complexity is in order to obtain the full Carleman vector. The quantity  $\mathcal{R}$  corresponds to an inverse amplitude for obtaining the state at the final time, so tells us how many steps of amplitude amplification are needed in the algorithm. In practice, we want only  $\mathbf{u}(T)$  rather than the full vector. That implies a further factor in the complexity of  $\|\mathbf{y}(T)\|/\|\mathbf{u}(T)\|$ , corresponding to the inverse amplitude for obtaining the component of the Carleman vector containing the solution. That gives a



factor in the complexity of

$$\frac{\|\mathbf{y}(T)\|}{\|\mathbf{u}(T)\|} \mathcal{R} \geq \left[ \frac{(1 - \|\mathbf{u}_{\text{in}}\|^{2N})}{(1 - \|\mathbf{u}_{\text{in}}\|^2)} \right]^{1/2} \frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}. \quad (62)$$

From the equation above we can see how  $\mathcal{R}$  grows exponentially in  $N$  for  $\|\mathbf{u}_{\text{in}}\| > 1$ .

Now with the rescaling, we simply divide each  $\mathbf{u}_{\text{in}}$  or  $\mathbf{u}(T)$  by  $\gamma$ . That gives us

$$\frac{\|\tilde{\mathbf{y}}(T)\|}{\|\tilde{\mathbf{u}}(T)\|} \tilde{\mathcal{R}} \geq \left[ \frac{(1 - \|\mathbf{u}_{\text{in}}\|^{2N}/\gamma^{2N})}{(1 - \|\mathbf{u}_{\text{in}}\|^2/\gamma^2)} \right]^{1/2} \frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}. \quad (63)$$

With the choice  $\gamma^{M-1} = |\lambda_0|/\|F_M\|$ , we obtain

$$\frac{\|\tilde{\mathbf{y}}(T)\|}{\|\tilde{\mathbf{u}}(T)\|} \tilde{\mathcal{R}} \geq \frac{1}{\sqrt{1 - R^{2/(M-1)}}} \frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}. \quad (64)$$

We then can see that the amplitude amplification cost can be exponentially reduced when  $\|\mathbf{u}_{\text{in}}\| > 1$ . We could also use  $\gamma = \|\mathbf{u}_{\text{in}}\|$  to give

$$\frac{\|\tilde{\mathbf{y}}(T)\|}{\|\tilde{\mathbf{u}}(T)\|} \tilde{\mathcal{R}} \geq \sqrt{N} \frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|}, \quad (65)$$

but that bound is looser for realistic parameters.

A further consideration is the relation between the relative error in the solution for  $\tilde{\mathbf{y}}(T)$  and that for  $\mathbf{u}(T)$ . The complexity of the solution for the ODE solver is in terms of the former, whereas we need to bound the relative error in  $\mathbf{u}(T)$ . We have the error upper bounded by (with hats used to indicate results given by the linear equation solver)

$$\begin{aligned} \|\hat{\mathbf{u}}(T) - \mathbf{u}(T)\| &\leq \gamma \|\hat{\mathbf{y}}(T) - \mathbf{y}(T)\| \\ &\leq \gamma \epsilon y_{\text{max}} \\ &\leq \gamma \left[ \frac{(1 - \|\mathbf{u}_{\text{in}}\|^{2N}/\gamma^{2N})}{(1 - \|\mathbf{u}_{\text{in}}\|^2/\gamma^2)} \right]^{1/2} \frac{\|\mathbf{u}_{\text{in}}\|}{\gamma} \\ &\leq \epsilon \|\mathbf{u}_{\text{in}}\| \frac{1}{\sqrt{1 - R^{2/(M-1)}}}. \end{aligned} \quad (66)$$

In the second line we have assumed that the ODE solver has given the solution for  $\mathbf{y}(T)$  to within error  $\epsilon y_{\text{max}}$ . This shows that the relative error in  $\mathbf{u}(T)$  is the same as that for  $\tilde{\mathbf{y}}(T)$ , up to a factor of  $1/\sqrt{1 - R^{2/(M-1)}}$  which should be close to 1. We can also use the simpler but looser upper bound

$$\|\hat{\mathbf{u}}(T) - \mathbf{u}(T)\| \leq \epsilon \|\mathbf{u}_{\text{in}}\| \sqrt{N}, \quad (67)$$

which is obtained by noting that the expression in the square brackets in the third line of Eq. (66) is upper bounded by  $N$ .

We can now use the ODE solver given in ref. 3 in combination with our rescaling technique to provide our quantum algorithm for Problem 1.

**Lemma 5. (Complexity of solving ODE)** There is a algorithm to solve the nonlinear ODE from Eq. (3) i.e., to produce a quantum state  $|\hat{\mathbf{u}}(T)\rangle$  encoding the solution such that  $\|\hat{\mathbf{u}}(T) - \mathbf{u}(T)\| \leq \epsilon \|\mathbf{u}_{\text{in}}\|$ , using an average number

$$\mathcal{O}\left(\frac{1}{\sqrt{1 - R^{2/(M-1)}}} \frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|} \lambda_{F_1} TN \log\left(\frac{N}{\epsilon}\right) \log\left(\frac{N\lambda_{F_1} T}{\epsilon}\right)\right), \quad (68)$$

of calls to oracles for  $F_1$  and  $F_M$ ,

$$\mathcal{O}\left(\frac{1}{\sqrt{1 - R^{2/(M-1)}}} \frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|} \lambda_{F_1} TN^2 \log\left(\frac{N}{\epsilon}\right)\right), \quad (69)$$

calls to oracles for preparation of  $\mathbf{u}_{\text{in}}$ , and

$$\mathcal{O}\left(\frac{1}{\sqrt{1 - R^{2/(M-1)}}} \frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|} \lambda_{F_1} TN^2 M \log\left(\frac{N}{\epsilon}\right) \log\left(\frac{N\lambda_{F_1} T}{\epsilon}\right)^2 \log n\right), \quad (70)$$

additional gates for dimension  $n$ , with

$$N = \mathcal{O}\left((M-1) \frac{\log(1/\epsilon)}{\log(1/R)}\right). \quad (71)$$

We require that  $R < 1$  and assume that  $\lambda_{F_M}/\|F_M\| = \mathcal{O}(\lambda_{F_1}/\|F_1\|)$  for the block encodings of  $F_1$  and  $F_M$ .

**Proof.** The main step to derive our quantum algorithm is first to apply the Carleman linearisation in the rescaled nonlinear ODE problem, which is given in Eq. (26). We then have a linear ODE problem with the Carleman matrix of order  $N$ , denoted  $\tilde{\mathcal{A}}_N$ . We can then apply the ODE solver given in ref. 3 to this equation.

There are then a number of considerations needed to give the overall complexity.

- We need to multiply by a further factor of  $\|\tilde{\mathbf{y}}(T)\| / \|\tilde{\mathbf{u}}(T)\|$  to obtain the correct component of the solution containing the approximation of  $\mathbf{u}(T)$ . The product of that with  $\tilde{\mathcal{R}}$  is given above in Eq. (64).
- The value of  $\lambda_{\tilde{\mathcal{A}}_N}^{-1}$  is given above in Eq. (58) under the assumption  $\lambda_{F_M}/\|F_M\| = \mathcal{O}(\lambda_{F_1}/\|F_1\|)$ , which gives  $\lambda_{\tilde{\mathcal{A}}_N}^{-1} = \mathcal{O}(N\lambda_{F_1})$ .
- The matrix  $\tilde{\mathcal{A}}_N$  can be block encoded with  $\mathcal{O}(1)$  calls to the oracles for  $F_1, F_M$ . There is an extra  $\mathcal{O}(N)$  factor for the number of calls to  $\mathbf{u}_{\text{in}}$ . The implementation of the oracles is explained in the Supplementary Section VII of the Supplementary Information.
- The choice of the Carleman order  $N$  in order to obtain a sufficiently accurate solution is given in Eq. (40). The error from the Carleman truncation can be chosen to be a fraction of the total allowable relative error  $\epsilon$  here, which is accounted for using the order notation for  $N$ .
- The solution for the ODE can be given to relative error  $\epsilon/\sqrt{N}$ . According to Eq. (67) that will ensure that the relative error in  $\mathbf{u}(T)$  obtained is  $\epsilon$  as required. It is for this reason that we have replaced the  $1/\epsilon$  in the complexity for the ODE solver with  $N/\epsilon$ . For the additional elementary gates, the block encoding is as in Supplementary Section VII of the Supplementary Information requires a factor of  $\mathcal{O}(NM \log n)$  for swapping target registers into the appropriate location. That is a factor on the number of block encodings of  $\tilde{\mathcal{A}}_N$ . Moreover, ref. 3 gives a log factor to account for the complexity of correctly giving the weighting in the Taylor series. For simplicity we give the product of these factors, but these factors are for different contributions to the complexity and we could instead give a more complicated expression with the maximum of  $NM \log n$  and the logarithm.

We can compare our quantum algorithm performance with what is given in Theorem 8 of ref. 7 for the case  $M = 2$ . The complexity given in that theorem can be simplified to the situation we consider by removing the driving term and replacing  $\|A\|$  with  $\lambda_{\tilde{\mathcal{A}}_N}$ . Then the complexity from ref. 7 is

$$\mathcal{O}\left(\frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|} \lambda_{F_1} TN \text{poly}\left(N, \log\left(\frac{1}{\epsilon}\right), \log(TN\lambda_{F_1})\right)\right). \quad (72)$$

The speedup is unclear because the complexity in that work is given in terms of poly factors. That work appears to be assuming a rescaling in order to avoid complexity exponential in  $N$ , but by assuming  $\|\mathbf{u}_{\text{in}}\| = 1$ . The problem is that they give a formula for  $N$  as

$$N = \left\lceil \frac{2 \log(T \|F_2\| / \delta \|\mathbf{u}(T)\|)}{\log(1 / \|\mathbf{u}_{\text{in}}\|)} \right\rceil. \quad (73)$$

Using  $\|\mathbf{u}_{\text{in}}\| = 1$  in that formula gives infinite  $N$ . In contrast, here we have given the rescaling explicitly and given a working formula for  $N$ .

### Application to the quantum nonlinear PDE problem

**Complexity of the quantum algorithm.** We now demonstrate our techniques applied to the nonlinear PDE<sup>23</sup>

$$\partial_t u(\mathbf{x}, t) = D \Delta u(\mathbf{x}, t) + cu(\mathbf{x}, t) + bu^M(\mathbf{x}, t), \quad (74)$$

for some diffusion coefficient  $D \geq 0$  and constants  $c, b \in \mathbb{R}$ . As a simple means of discretisation we consider finite differences with periodic boundary conditions, which leads to a vector-valued ODE that approximates the dynamics in Eq. (74). We go beyond the two-point stencil demonstrated in ref. 23 and apply higher-order finite differences similar to ref. 13 for the linear case.

We discretise a  $d$ -dimensional space in each direction with uniformly equidistant grid points. As a result, we obtain a nonlinear system of ODEs as in Eq. (3) with  $n$  grid points in total, or  $n^{1/d}$  in each direction. Moreover, we consider the width of the simulation region to be 1 in each direction, so  $x_j \in [0, 1]$ , for simplicity.

The linear operator  $F_1$  resulting from the spatial discretisation of our PDE is given by

$$F_1 = DL_{k,d} + c\mathbb{I}^{\otimes d}, \quad (75)$$

where  $\mathbb{I}$  is the  $n^{1/d} \times n^{1/d}$  identity matrix, and

$$L_{k,d} = \sum_{\mu=1}^d \mathbb{I}^{\otimes(\mu-1)} \otimes L_k \otimes \mathbb{I}^{\otimes(d-\mu)}. \quad (76)$$

The operator  $L_{k,d}$  above for the discretised Laplacian in dimension  $d$  is constructed from the sum of the discretised Laplacians in one dimension,  $L_k$ . Here,  $k$  is the order, so the truncation error scales as the inverse grid spacing to the power of  $2k - 1$ , and  $2k + 1$  stencil points are used.

A Laplacian in one dimension with a  $k$ th order approximation and periodic boundary conditions can be expressed in terms of weights  $a_j$  as (see ref. 13)

$$L_k = n^{2/d} \left( a_0 I + \sum_{j=1}^k a_j (S^j + S^{-j}) \right), \quad (77)$$

where  $S$  is a  $n^{1/d} \times n^{1/d}$  matrix, where the entries are  $S_{i,j} = \delta_{i,j+1 \bmod n^{1/d}}$ ;  $S$  is also known as a circulant matrix. Note that for a total of  $n^{1/d}$  grid points and a region size of 1 in each direction, the grid spacing is  $1/n^{1/d}$ . The method to obtain the coefficients  $a_j$  for the Laplacian operator is given in Supplementary Section V of the Supplementary Information guarantees that

$$a_0 + 2 \sum_{j=1}^k a_j = 0, \quad (78)$$

and we provide the coefficients for  $1 \leq k \leq 5$  in Table 2 (these are from ref. 28). Moreover, this procedure leads to a truncation error in the

representation of the Laplacian operator which scales as (for the 2-norm)<sup>13,29</sup>

$$\mathcal{O} \left( C(u, k) \sqrt{n} \left( \frac{e}{2} \right)^{2k} n^{-(2k+1)/d} \right), \quad (79)$$

where  $C(u, k)$  is a constant depending on the  $(2k + 1)$ st spatial derivative in each direction

$$C(u, k) = \sum_{j=1}^d \left| \frac{d^{2k+1} u}{dx_j^{2k+1}} \right|. \quad (80)$$

This expression is obtained from that in refs. 13,29 by adding the errors for derivatives in each direction.

We also have the matrix  $F_M$  resulting from the spatial discretisation of the nonlinear part  $bu^M(\mathbf{x}, t)$  that is a rectangular matrix  $F_M$ ,

$$F_M : \mathbb{R}^{n^M} \rightarrow \mathbb{R}^n, \quad (81)$$

operating on the vector  $\mathbf{u}^{\otimes M}$ , as given in Eq. (10). Since we are only interested in the components  $u_i^M$  from  $\mathbf{u}^{\otimes M}$ , where  $i = 1, 2, \dots, n$ ,  $F_M$  is a one sparse matrix with the non-zero components given by  $b$ . Hence  $\|F_M\| = \|F_M\|_{\max} = |b|$ . In the case  $M = 2$ , where

$$\mathbf{u}^{\otimes 2} = (u_1^2, u_1 u_2, \dots, u_1 u_n, u_2^2, \dots, u_n u_{n-1}, u_n^2)^T \in \mathbb{R}^{n^2}, \quad (82)$$

we can express  $F_2$  as

$$[F_2]_{pq} = \begin{cases} b, & q = p + (p-1)n \\ 0, & \text{otherwise.} \end{cases} \quad (83)$$

Returning to the Laplacian operator with periodic boundary conditions, we see that Eq. (77) is a circulant matrix, so its eigenvalues are given by

$$\begin{aligned} \lambda_\ell(L_k) &= n^{2/d} \left[ a_0 + \sum_{j=1}^k a_j (\omega^{\ell j} + \omega^{-\ell j}) \right] \\ &= n^{2/d} \left[ a_0 + 2 \sum_{j=1}^k a_j \cos \left( \frac{2\pi \ell j}{n^{1/d}} \right) \right], \quad \ell \in [n^{1/d} - 1] \end{aligned} \quad (84)$$

where  $\omega = e^{i2\pi/n^{1/d}}$ . Since the  $a_j$ , with  $j = 0, 1, \dots, k$ , satisfy the condition in Eq. (78), we see that for  $\ell = 0$ ,  $\lambda_0 = 0$  gives the maximum eigenvalue and  $\lambda_\ell < 0$  for  $\ell \neq 0$ . Moreover, using the triangle inequality in Eq. (77) we see that

$$\|L_k\| \leq n^{2/d} \left( |a_0| + 2 \sum_{j=1}^k |a_j| \right). \quad (85)$$

By a simple application of Gershgorin's circle theorem, one may obtain the asymptotic bound (Lemma 2 in ref. 13, Lemma 6 in ref. 29),

$$\|L_k\| \leq n^{2/d} \frac{4\pi^2}{3}. \quad (86)$$

From the eigenvalues of  $L_k$  we can then determine the eigenvalues of  $F_1$  (which is symmetric so equal to  $(F_1 + F_1^\dagger)/2$ ) as defined in Eq. (75) as

$$\lambda_{\rightarrow \ell}(F_1) = c + Dn^{2/d} \sum_{p=1}^d \left[ a_0 + 2 \sum_{j=1}^k a_j \cos \left( \frac{2\pi \ell_p j}{n^{1/d}} \right) \right]. \quad (87)$$

As discussed above the maximum eigenvalue of  $L_k$  is 0 with periodic boundary conditions (it can be negative for non-periodic boundary conditions). For Carleman linearisation to be successful, we require  $R < 1$  and, in particular,  $\lambda_0 < 0$ . Since the maximum eigenvalue of  $F_1$  is  $c$ , we choose negative  $c$  such that the overall dynamics becomes dissipative and satisfies

**Table 2 | Central finite difference coefficients for approximating a second derivative in one dimension<sup>28</sup>**

Order $k$	Coefficients $a_0$ to $a_k$
1	$-2, 1$
2	$-5/2, 4/3, -1/12$
3	$-49/18, 3/2, -3/20, 1/90$
4	$-205/72, 8/5, -1/5, 8/315, -1/560$
5	$-5269/1800, 5/3, -5/21, 5/126, -5/1008, 1/3150$

the stability condition  $R < 1$ . Moreover, we obtain the following bounds

$$\|F_1\| \leq |c| + dDn^{2/d} \left( |a_0| + \sum_{j=1}^k |a_j| \right) \leq |c| + dDn^{2/d} \frac{4\pi^2}{3}. \quad (88)$$

The value of  $\lambda_{F_1}$  for the block encoding of  $F_1$  can be determined in a similar way. The block encoding can be implemented by a linear combination of unitaries of the identity and powers of the circulant matrices  $S$ . The value of  $\lambda_{F_1}$  is then exactly equal to the sum

$$\lambda_{F_1} = |c| + dDn^{2/d} \left( |a_0| + \sum_{j=1}^k |a_j| \right) \leq |c| + dDn^{2/d} \frac{4\pi^2}{3}. \quad (89)$$

Similarly, since  $F_M$  is one-sparse it can be easily block encoded with a value of  $\lambda_{F_M}$  equal to its norm of  $|b|$ . We then obtain the value of  $\lambda$  for the complete block encoding as

$$\begin{aligned} \lambda_{\mathcal{A}_N} &= N\lambda_{F_1} + (N - M + 1)\gamma^{M-1}\lambda_{F_M} \\ &\leq N(|c| + dDn^{2/d} \frac{4\pi^2}{3}) + (N - M + 1)\gamma^{M-1}|b|. \end{aligned} \quad (90)$$

The condition on the dissipativity of the ODE  $R < 1$  implies that

$$N \left( |c| + dDn^{2/d} \frac{4\pi^2}{3} \right) > (N - M + 1)\gamma^{M-1}|b|. \quad (91)$$

Given these results for the discretisation of the PDE, we can use Lemma 5 to provide the following corollary.

**Corollary 6. (Complexity of solving a dissipative reaction-diffusion PDE)** There is a quantum algorithm to solve the nonlinear PDE in Eq. (74) i.e., to produce a quantum state  $|\hat{\mathbf{u}}(T)\rangle$  encoding the solution such that  $\|\hat{\mathbf{u}}(T) - \mathbf{u}(T)\| \leq \varepsilon \|\mathbf{u}_{\text{in}}\|$ , using an average number

$$\mathcal{O} \left( \frac{1}{\sqrt{1 - R^{2/(M-1)}}} \frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|} (dDn^{2/d} + |c|)TN \log \left( \frac{N}{\varepsilon} \right) \log \left( \frac{N(dDn^{2/d} + |c|)T}{\varepsilon} \right) \right), \quad (92)$$

of calls to oracles for  $F_1$  and  $F_M$ , as defined in Eq. (75) and Eq. (83) respectively,

$$\mathcal{O} \left( \frac{1}{\sqrt{1 - R^{2/(M-1)}}} \frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|} (dDn^{2/d} + |c|)TN^2 \log \left( \frac{N}{\varepsilon} \right) \right), \quad (93)$$

calls to oracles for preparation of  $\mathbf{u}_{\text{in}}$ , and

$$\mathcal{O} \left( \frac{1}{\sqrt{1 - R^{2/(M-1)}}} \frac{\|\mathbf{u}_{\text{in}}\|}{\|\mathbf{u}(T)\|} (dDn^{2/d} + |c|)TN^2M \log \left( \frac{N}{\varepsilon} \right) \log \left( \frac{N(dDn^{2/d} + |c|)T}{\varepsilon} \right)^2 \log n \right), \quad (94)$$

additional gates, with

$$N = \mathcal{O} \left( (M - 1) \frac{\log(1/\varepsilon)}{\log(1/R)} \right). \quad (95)$$

We require that  $R < 1$ , where  $R$  is computed from the discretised input vector  $\mathbf{u}_{\text{in}}$ .

**Proof.** We first discretise the reaction-diffusion problem in Eq. (74) to the nonlinear ODE system with  $n$  discretisation points. We consider just the error in the solution of this ODE here, with the choice of  $n$  to accurately approximate the solution of the PDE described below. For this ODE we have an explicit bound for  $\lambda_{F_1}$  given in Eq. (89), and can use it in the expressions in Lemma 5. For this simple  $F_M$  the values of  $\lambda_{F_M}$  and  $\|F_M\|$  are equal, so the condition  $\lambda_{F_M}/\|F_M\| = \mathcal{O}(\lambda_{F_1}/\|F_1\|)$  is satisfied.

Note that for this result the oracles for  $F_1$  and  $F_M$  can be easily implemented in terms of calls to elementary gates, with logarithmic complexity in  $n$  and linear complexity in  $M$ . Powers of the circulant matrices can be implemented with modular addition, and  $F_M$  can be implemented via equality tests between the copies it acts upon. Note also that, apart from the  $R < 1$  condition, this complexity scales as  $n^{2/d}$  up to logarithmic factors. For  $d \geq 3$  this complexity is sublinear in  $n$ . This factor comes from the size of the discretised Laplacian, and is similar to that for quantum algorithms for linear PDEs. If we had the factor of  $\|\mathbf{u}_{\text{in}}\|^{2N}$  as in ref. 23, then because  $\|\mathbf{u}_{\text{in}}\|^2 \propto n$  with the discretisation there would be a further factor of  $n^N$  for the scaling with  $n$ , making the complexity far worse than that for a simple classical solver.

**Stability and discretisation.** Here we discuss conditions on nonlinear differential equations of the type in Eq. (74) so that numerical schemes based on Carleman linearisation are stable. Recall that for the ODE we have the stability condition  $R < 1$  with

$$R = \frac{\|F_M\| \cdot \|\mathbf{u}_{\text{in}}\|^{M-1}}{|\lambda_0|}. \quad (96)$$

That condition is not ideal here, because the 2-norm of the solution increases with the number of discretisation points. Thus this condition for the stability depends not only on the underlying PDE and initial state but on its discretisation.

Ideally we would aim for a condition on the max-norm of the solution. That can then be used in order to guarantee stability of the solution as well as to bound error. For example, ref. 23 considers stability in their Lemma 2.1 and bounds error in their Theorem 3.3. A simple stability criterion can be given as

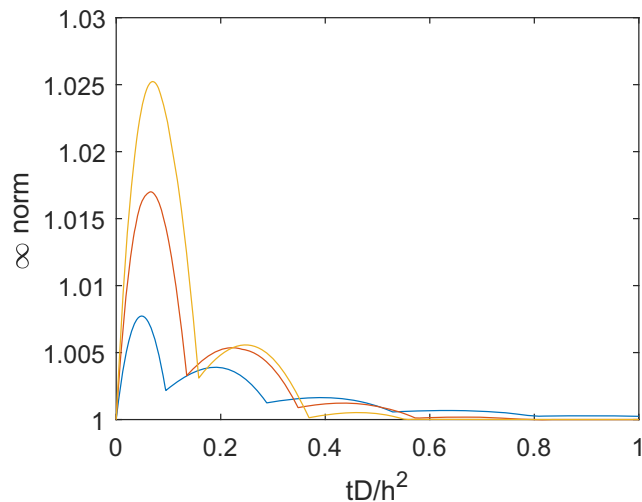
$$\|\mathbf{u}_{\text{in}}\|_{\max}^{M-1} \frac{b}{|c|} < 1. \quad (97)$$

Before discretisation, the stability can be shown simply by considering the infinitesimal time interval  $dt$  and using

$$\begin{aligned} &\|u(\mathbf{x}, t) + dt[D\Delta u(\mathbf{x}, t) + cu(\mathbf{x}, t) + bu^M(\mathbf{x}, t)]\|_{\max} \\ &= \|(\mathbb{I} + dt D\Delta)\{u(\mathbf{x}, t) + dt[cu(\mathbf{x}, t) + bu^M(\mathbf{x}, t)]\}\|_{\max} \\ &\leq \|\mathbb{I} + dt D\Delta\|_{\infty} \|u(\mathbf{x}, t) + dt[cu(\mathbf{x}, t) + bu^M(\mathbf{x}, t)]\|_{\max}. \end{aligned} \quad (98)$$

Now using the triangle inequality

$$\begin{aligned} \|u(\mathbf{x}, t) + dt[cu(\mathbf{x}, t) + bu^M(\mathbf{x}, t)]\|_{\max} &\leq \|u(\mathbf{x}, t) + dt cu(\mathbf{x}, t)\|_{\max} + dt \|bu^M(\mathbf{x}, t)\|_{\max} \\ &= (1 + dt c) \|u(\mathbf{x}, t)\|_{\max} + dt b \|u(\mathbf{x}, t)\|_{\max}^M. \end{aligned} \quad (99)$$



**Fig. 3** | The (induced)  $\infty$ -norm of the exponential of  $tD\Delta_h$ , when using a discretisation of the Laplacian of second order (blue), third order (red), and fourth order (orange).

Then if  $(b/c) \|u(\mathbf{x}, t)\|_{\max}^{M-1} \leq 1$  this expression is upper bounded by  $\|u(\mathbf{x}, t)\|_{\max}$ .

Moreover, it is a standard result that  $\|\mathbb{I} + dt D\Delta\|_{\infty} = 1$ . That is, the diffusion equation smooths out any peaks in the distribution. That expression also holds if we consider the discretised form, but only using the first-order discretisation. Then for spatial grid spacing  $h$ , the discretised form in one dimension has  $1 - 2dt D/h^2$  on the diagonal, and  $dt D/h^2$  on the two off-diagonals. The sum of the absolute values along a row for this matrix is then exactly 1, giving an  $\infty$ -norm of 1. That means Eq. (98) implies  $\|u(\mathbf{x}, t)\|_{\max}$  is non-increasing for the PDE given the stability criterion in Eq. (97).

This result for the  $\infty$ -norm no longer holds for the discretised PDE when using higher-order discretisations. For example, for the second-order discretisation, the  $-1/12$  on the off-diagonals means that

$$\|\mathbb{I} + dt DL_2\|_{\infty} = 1 + dt D/(3h^2). \quad (100)$$

That means that the max-norm is only upper bounded by the initial max-norm multiplied by a factor of  $\exp(D/(3h^2))$ . In practice it is found that the max-norm is far better behaved. If we calculate the  $\infty$ -norm of  $\exp(tDL_k)$ , then we obtain the results shown in Fig. 3. For the second-order discretisation the initial slope is  $1/3$  as predicted using infinitesimal  $t$ , but the peak value is less than 1% above 1. For the higher-order discretisations this maximum increases, but it is still small for these orders. Therefore we find that if we consider the max-norm for just evolution under the discretised Laplacian then it is well-behaved, but that does not imply the result for the nonlinear discretised PDE.

To determine stability for the 2-norm, the equivalent of Eq. (98) gives

$$\begin{aligned} \|u(\mathbf{x}, t + dt)\| &\leq \|\mathbb{I} + dt D\Delta\| \times \|u(\mathbf{x}, t) + dt[cu(\mathbf{x}, t) + bu^M(\mathbf{x}, t)]\| \\ &\leq \|u(\mathbf{x}, t) + dt[cu(\mathbf{x}, t) + bu^M(\mathbf{x}, t)]\| \\ &\leq \|u(\mathbf{x}, t)\| + dt[c\|u(\mathbf{x}, t)\| + b\|u^M(\mathbf{x}, t)\|] \\ &\leq \|u(\mathbf{x}, t)\| + dt[c\|u(\mathbf{x}, t)\| + b\|u(\mathbf{x}, t)\|_{\max}^{M-1}\|u(\mathbf{x}, t)\|]. \end{aligned} \quad (101)$$

Therefore the 2-norm is non-increasing provided  $(b/c) \|u(\mathbf{x}, t)\|_{\max}^{M-1} \leq 1$ . In the discretised case the non-positive eigenvalues of the discretised Laplacian mean that the 2-norm is still stable given this condition, though as noted above it is possible for  $\|u\|_{\max}$  to increase above its initial value.

However, for the purpose of solving the ODE using Carleman linearisation, what matters is not the stability of the nonlinear equation, but that of the linearised equation. That is because the stability of the linearised

equation governs the condition number of the linear equations to solve, and in turn that is proportional to the complexity. For example, in ref. 23 their Problem 1 assumes that  $\|F_M\| \leq |\lambda_0|$  ( $\lambda_1$  in the notation of that work) after some possible rescaling of the equation. Then Eq. (4.15) of that work gives  $\|\mathbb{I} + A/h\| \leq 1$  (with  $h$  the time discretisation), using that condition from Problem 1. That is then used to provide the bound on the norm of  $\|L^{-1}\|$  in Eq. (4.28) of that work, which is then used to give the bound on the condition number proportional to the number of time steps in Eq. (4.29) in ref. 23.

According to our analysis above, the stability of the linearised system will be satisfied provided  $\gamma^{M-1} \leq |\lambda_0|/\|F_M\|$ . For the discretised PDE here we have  $\lambda_0 = c$  and  $\|F_M\| = b$ . That means if  $\gamma \leq \|u_{\text{in}}\|_{\max}$ , then the stability condition in Eq. (97) implies the stability of the matrix after Carleman linearisation. That condition is needed in order to be able to use the ODE solver of Ref. 3, but it will mean that the rescaling gives a smaller probability of success for obtaining the correct component of the Carleman vector than if we had the stability condition  $R < 1$ .

However, if we have sufficiently small  $b/|c|$ , then the condition  $R < 1$  would be satisfied, so

$$\|u_{\text{in}}\|^{M-1} \frac{b}{|c|} < 1. \quad (102)$$

Because  $\|u_{\text{in}}\|$  increases with the number of discretisation points as  $\sqrt{n}$ , this inequality can only be satisfied if the number of discretisation points is made as small as possible. This gives a strong motivation for using the higher-order spatial discretisation of the PDE. See Supplementary Section IV of the Supplementary Information for discussion of the number of points needed.

**Error analysis.** The overall error  $\varepsilon$  comes from three different parts,

- the spatial discretisation error of the semi-discrete dynamics  $\varepsilon_{\text{disc}}$
- the error  $\varepsilon_{\text{Carl}}$  contributed by truncation in the Carleman linearisation as bounded in Lemma 4, and
- the error in the time evolution  $\varepsilon_{\text{time}}$  due to the Taylor series, as introduced earlier in the treatment of the linearised differential equation solver. As usual in this type of analysis, we can simplify the discussion by taking the error to be  $\varepsilon$  for each of these contributions. In reality, the contribution to the error from each source would need to be taken to be a fraction of  $\varepsilon$  (e.g.  $\varepsilon/3$ ), but because that fraction would at most give a constant factor to the complexity, it would not affect the complexities quoted using  $\mathcal{O}$ .

We have already considered  $\varepsilon_{\text{time}}$  and  $\varepsilon_{\text{Carl}}$  above in Corollary 6. The time discretisation error will not be further considered here, but we will discuss how the Carleman error can be alternatively bounded in situations where the PDE is stable but  $R \geq 1$ . Above we show that the ODE needs  $R < 1$  for the quantum solution to be efficient, but this bound on the Carleman error will be useful if that limitation can be circumvented.

In the case of higher-order discretised Laplacians we obtain a somewhat worse bound as derived in Supplementary Section II C of the Supplementary Information,

$$\| \eta_j(t) \|_{\max} \lesssim G_{\kappa}^{dj} \left( \frac{\|F_M\|_{\infty}}{|c|} \|u_{\text{in}}\|_{\max}^{M-1} G_{\kappa}^{dM} \right)^k f_{j,k,M}(|c|t), \quad (103)$$

where

$$G_{\kappa} := \max_{\tau \geq 0} \|e^{L_{\kappa}\tau}\|_{\infty}. \quad (104)$$

The  $\lesssim$  is because it is assuming that the max-norm of the solution is not increasing. We can use  $\leq$  if  $\|u_{\text{in}}\|_{\max}$  is replaced with the maximum of  $\|u\|_{\max}$  over time. The quantity  $G_{\kappa}$  is greater than 1 for higher-order discretised Laplacians, so this is a slightly larger upper bound than in the case of first-order discretised Laplacians. Nevertheless, the Carleman error may



be made arbitrarily small with order provided

$$\|\mathbf{u}_{\text{in}}\|_{\text{max}}^{M-1} \frac{b}{|c|} G_{\kappa}^{dM} < 1. \quad (105)$$

This condition is slightly stronger than the condition for stability of the PDE by the factor of  $G_{\kappa}^{dM}$ , but that will typically be close to 1. Typically this will be a much weaker requirement than the stability condition  $R < 1$ .

Next, we consider the bound on the error due to the spatial discretisation, which can be used to derive the appropriate number of discretisation points  $n$  to use. Using that in Corollary 6 then gives the complexity entirely in terms of the parameters of the problem instead of the value chosen for  $n$ . Our bound on the error is as given in the following Lemma, with the proof given in Supplementary Section III of the Supplementary material.

**Lemma 7. (Nonlinear PDE solution error when discretising the Laplacian with higher-order finite differences).** Using a higher-order finite difference discretisation with  $2k + 1$  stencil points in each direction, the solution of the PDE

$$\frac{d\mathbf{u}}{dt} = (DL_{k,d} + c)\mathbf{u} + b\mathbf{u}^{\otimes M}, \quad (106)$$

at time  $T > 0$  has error due to spatial discretisation when  $c < 0$  and  $|c| > M|b|\|\mathbf{u}_{\text{in}}\|_{\text{max}}^{M-1}$  bounded as

$$\|\varepsilon_{\text{disc}}(T)\| = O\left(C(u, k)\sqrt{n}\left(\frac{\varepsilon}{2}\right)^{2k} n^{-(2k-1)/d} \frac{1 - \exp\left\{\left(c + M|b|\|\mathbf{u}_{\text{in}}\|_{\text{max}}^{M-1}\right)t\right\}}{|c + M|b|\|\mathbf{u}_{\text{in}}\|_{\text{max}}^{M-1}|}\right), \quad (107)$$

where  $C(u, k)$  given in Eq. (80) is a constant depending on the  $(2k + 1)$ st spatial derivative of the solution assuming sufficient regularity,  $n$  is the number of grid points used, and  $d$  is the number of dimensions.

In this result, we are considering the continuous time evolution. Note that for the stability of the discretisation error we use the condition  $|c| > M|b|\|\mathbf{u}_{\text{in}}\|_{\text{max}}^{M-1}$ , which is stronger than the condition  $|c| > |b|\|\mathbf{u}_{\text{in}}\|_{\text{max}}^{M-1}$  for PDE stability. This appears to be a fundamental condition due to the nonlinearity, because the derivative of the order- $M$  nonlinearity produces a factor of  $M$ .

Next, if we take  $\varepsilon_{\text{disc}} \propto \varepsilon$ , then solving Eq. (107) for  $n$  gives

$$n = \Omega\left(\left[\frac{C(u, k)\left(\frac{\varepsilon}{2}\right)^{2k}}{|c + M|b|\|\mathbf{u}_{\text{in}}\|_{\text{max}}^{M-1}|}\right]^{\frac{2d}{2(2k-1)+d}}\right). \quad (108)$$

That is, this choice of  $n$  is sufficient to give  $\varepsilon_{\text{disc}}$  as some set fraction of  $\varepsilon$ . In practice we would choose the minimum  $n$  needed to give the desired accuracy, so we would choose  $n$  proportional to the expression in Eq. (108). For simplicity of the solution for  $n$ , we have used

$$1 - \exp\left\{\left(c + M|b|\|\mathbf{u}_{\text{in}}\|_{\text{max}}^{M-1}\right)t\right\} \leq 1. \quad (109)$$

In Supplementary Section IV of the Supplementary Information we show how the discretisation error is reduced with the number of grid points with different orders of discretisation and a simple toy model.

The benefit of having fewer grid points comes with the drawback of having a less sparse operator. The block encoding of  $F_1$  is performed by a linear combination of unitaries over  $(2k + 1)$  basis states with amplitudes given by a table, and so has complexity in terms of elementary gates proportional to  $k$ . That is not immediately obvious from Corollary 6, because it gives complexity in terms of block encodings of  $F_1$  and  $F_M$ . It is also possible for  $C(u, k)$  to increase with  $k$ . In a real implementation it would therefore be desirable to choose an optimal  $k$  to minimise the complexity, instead of taking  $k$  as large as possible.

## Data availability

No datasets were generated or analysed during the current study.

Received: 13 February 2025; Accepted: 25 July 2025;

Published online: 18 August 2025

## References

- Berry, D. W. High-order quantum algorithm for solving linear differential equations. *J. Phys. A Math. Theor.* **47**, 105301 (2014).
- Berry, D. W., Childs, A. M., Ostrander, A. & Wang, G. Quantum algorithm for linear differential equations with exponentially improved dependence on precision. *Commun. Math. Phys.* **356**, 1057–1081 (2017).
- Berry, D. W. & Costa, P. Quantum algorithm for time-dependent differential equations using Dyson series. *Quantum* **8**, 1369 (2024).
- Childs, A. M. & Liu, J.-P. Quantum spectral methods for differential equations. *Commun. Math. Phys.* **375**, 1427–1457 (2020).
- An, D., Liu, J.-P., Wang, D. & Zhao, Q. A theory of quantum differential equation solvers: limitations and fast-forwarding. *Commun. Math. Phys.* **406**, 189 (2025).
- Fang, D., Lin, L. & Tong, Y. Time-marching based quantum solvers for time-dependent linear differential equations. *Quantum* **7**, 955 (2023).
- Krovi, H. Improved quantum algorithms for linear and nonlinear differential equations. *Quantum* **7**, 913 (2023).
- An, D., Liu, J.-P. & Lin, L. Linear combination of hamiltonian simulation for nonunitary dynamics with optimal state preparation cost. *Phys. Rev. Lett.* **131**, 150603 (2023).
- An, D., Childs, A. M. & Lin, L. Quantum algorithm for linear non-unitary dynamics with near-optimal dependence on all parameters. Preprint at <https://arxiv.org/abs/2312.03916> (2023).
- Harrow, A. W., Hassidim, A. & Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **103**, 150502 (2009).
- Costa, P. C. et al. Optimal scaling quantum linear-systems solver via discrete adiabatic theorem. *PRX Quantum* **3**, 040303 (2022).
- Clader, B. D., Jacobs, B. C. & Sprouse, C. R. Preconditioned quantum linear system algorithm. *Phys. Rev. Lett.* **110**, 250504 (2013).
- Childs, A. M., Liu, J.-P. & Ostrander, A. High-precision quantum algorithms for partial differential equations. *Quantum* **5**, 574 (2021).
- Bagherimehrab, M., Nakaji, K., Wiebe, N. & Aspuru-Guzik, A. Fast quantum algorithm for differential equations. Preprint at <https://arxiv.org/abs/2306.11802> (2023).
- Jin, S., Liu, N. & Yu, Y. Quantum simulation of partial differential equations via Schrodingerisation. *Phys. Rev. Lett.* **133**, 230602 (2024).
- Jin, S., Liu, N. & Yu, Y. Quantum simulation of partial differential equations via Schrodingerisation: technical details. *Phys. Rev. A* **108**, 032603 (2023).
- Leyton, S. K. & Osborne, T. J. A quantum algorithm to solve nonlinear differential equations. Preprint at <https://arxiv.org/abs/0812.4423> (2008).
- Lloyd, S. et al. Quantum algorithm for nonlinear differential equations. Preprint at <http://arxiv.org/abs/2011.06571> (2020).
- Jin, S. & Liu, N. Quantum algorithms for computing observables of nonlinear partial differential equations. *Bulletin des Sciences Mathématiques* **194**, 103457 (2024).
- Jin, S., Liu, N. & Yu, Y. Time complexity analysis of quantum algorithms via linear representations for nonlinear ordinary and partial differential equations. *J. Comput. Phys.* **487**, 112149 (2023).
- Carleman, T. Application de la théorie des équations intégrales linéaires aux systèmes d'équations différentielles non linéaires. *Acta Math.* **59**, 63–87 (1932).
- Liu, J.-P. et al. Efficient quantum algorithm for dissipative nonlinear differential equations. *Proc. Natl Acad. Sci.* **118**, e2026805118 (2021).



23. Liu, J.-P. et al. Efficient quantum algorithm for nonlinear reaction-diffusion equations and energy estimation. *Commun. Math. Phys.* **404**, 963 (2023).
24. Li, X. et al. Potential quantum advantage for simulation of fluid dynamics. *Phys. Rev. Res.* **7**, 013036 (2025).
25. Xue, C., Xu, X.-F., Wu, Y.-C. & Guo, G.-P. Quantum algorithm for solving a quadratic nonlinear system of equations. *Phys. Rev. A* **106**, 032427 (2022).
26. Feingold, D. G. & Varga, R. S. Block diagonally dominant matrices and generalizations of the Gerschgorin circle theorem. *Pac. J. Math* **12**, 1241–1250 (1962).
27. van der Sluis, A. Gerschgorin domains for partitioned matrices. *Linear Algebra Appl.* **26**, 265–280 (1979).
28. Costa, P. C., Jordan, S. & Ostrander, A. Quantum algorithm for simulating the wave equation. *Phys. Rev. A* **99**, 012323 (2019).
29. Kivlichan, I. D., Wiebe, N., Babbush, R. & Aspuru-Guzik, A. Bounding the costs of quantum simulation of many-body physics in real space. *J. Phys. A Math. Theor.* **50**, 305301 (2017).

## Acknowledgements

M.E.S.M. and P.S. were supported by the Sydney Quantum Academy, Sydney, NSW, Australia. MESM was supported by the ARC Centre of Excellence for Quantum Computation and Communication Technology (CQC2T), project number CE170100012 and also supported by the Defense Advanced Research Projects Agency under Contract No.~HR001122C0074. D.W.B. worked on this project under a sponsored research agreement with Google Quantum AI. D.W.B. is also supported by Australian Research Council Discovery Projects DP190102633, DP210101367, and DP220101602.

## Author contributions

P.C.S.C. and D.W.B. jointly developed the algorithms, performed the complexity analysis, and co-wrote the manuscript. P.S. contributed to construction of the finite-difference coefficients and analysing sparsity of the Carleman matrix. M.E.S.M. wrote the section analysing the error with high-order finite difference discretisations. All authors participated in the discussion of results and writing and revision of the manuscript.

## Competing interests

D.W.B. is an Associate Editor for npj Quantum Information, but was not involved in the editorial review of, or the decision to publish this article. All other authors declare no competing interests.

## Additional information

**Supplementary information** The online version contains supplementary material available at <https://doi.org/10.1038/s41534-025-01084-z>.

**Correspondence** and requests for materials should be addressed to Pedro C. S. Costa.

**Reprints and permissions information** is available at <http://www.nature.com/reprints>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025