Efficient quantum algorithm for dissipative nonlinear differential equations

Andrew Childs
University of Maryland

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

UMIACS
University of Maryland
Institute for Advanced
Computer Studies

JOINT CENTER FOR
QUANTUM INFORMATION
AND COMPUTER SCIENCE


Jin-Peng Liu
UMD


Herman Kolden
NUST/MIT


Nuno Loureiro
MIT


Hari Krovi
Raytheon BBN


Konstantina Trivisa
UMD

arXiv:2011.03185

Quantum computation

By storing and processing information stored in quantum states, a quantum computer can solve certain problems dramatically faster than ordinary (“classical”) computers.

Quantum computation

By storing and processing information stored in quantum states, a quantum computer can solve certain problems dramatically faster than ordinary (“classical”) computers.

Exponential speedup: factoring, discrete log, computations in algebraic number fields, simulating quantum mechanics, approximating topological invariants, …

Quantum computation

By storing and processing information stored in quantum states, a quantum computer can solve certain problems dramatically faster than ordinary ("classical") computers.

Exponential speedup: factoring, discrete log, computations in algebraic number fields, simulating quantum mechanics, approximating topological invariants, …

Polynomial speedup: unstructured search, collision finding, graph properties, Boolean formula evaluation, NP-hard problems, …

Quantum computation

By storing and processing information stored in quantum states, a quantum computer can solve certain problems dramatically faster than ordinary ("classical") computers.

Exponential speedup: factoring, discrete log, computations in algebraic number fields, simulating quantum mechanics, approximating topological invariants, …

Polynomial speedup: unstructured search, collision finding, graph properties, Boolean formula evaluation, NP-hard problems, …

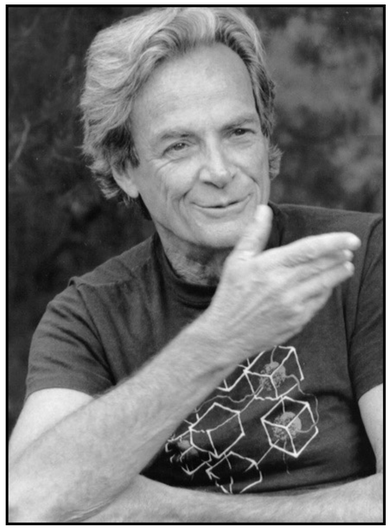
Quantum computers also provide a novel approach to problems in numerical analysis.

Main idea:

- Represent a vector $x \in \mathbb{C}^N$ by a quantum state $|x\rangle$ of $\log_2 N$ qubits.
- Show how to prepare such a state using $\mathrm{poly}(\log N)$ operations.
- Produces a quantum encoding of the solution. Less informative than an explicit solution, but much faster and still potentially useful.

Quantum simulation

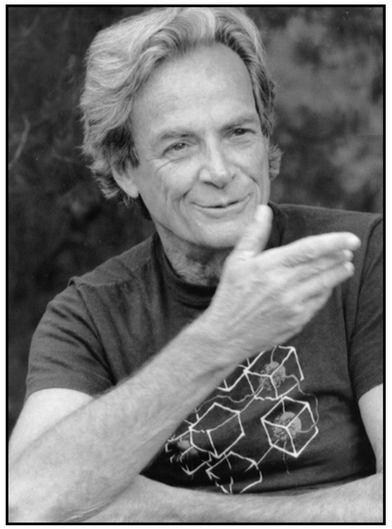
Quantum simulation



"… nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy."

Richard Feynman (1981)
Simulating physics with computers

Quantum simulation

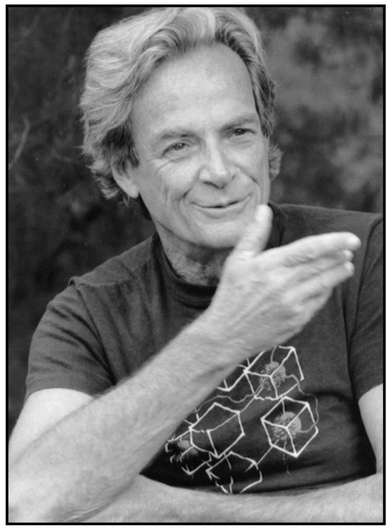


"… nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy."

Richard Feynman (1981)
Simulating physics with computers

Quantum simulation problem: Given a description of the Hamiltonian H, an evolution time t, and an initial state $|\psi(0)\rangle$, produce the final state $|\psi(t)\rangle$ (to within some error tolerance ϵ)

Quantum simulation



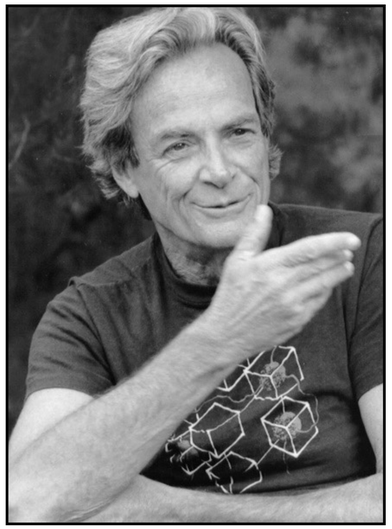
"… nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy."

Richard Feynman (1981)
Simulating physics with computers

Quantum simulation problem: Given a description of the Hamiltonian H, an evolution time t, and an initial state $|\psi(0)\rangle$, produce the final state $|\psi(t)\rangle$ (to within some error tolerance ϵ)

A classical computer cannot even represent the state efficiently

Quantum simulation



"… nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy."

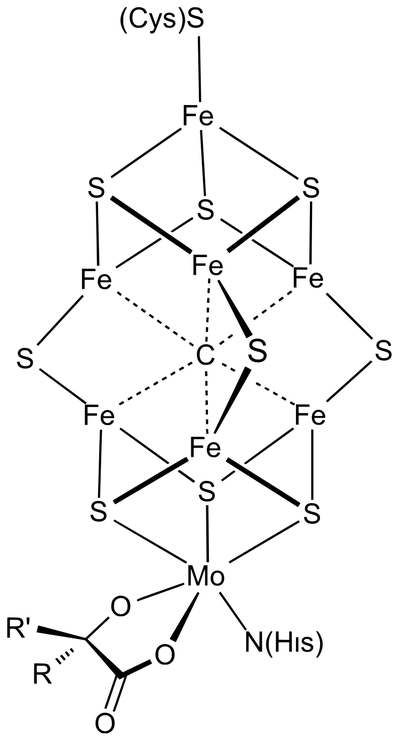
Richard Feynman (1981)
Simulating physics with computers

Quantum simulation problem: Given a description of the Hamiltonian H, an evolution time t, and an initial state $|\psi(0)\rangle$, produce the final state $|\psi(t)\rangle$ (to within some error tolerance ϵ)

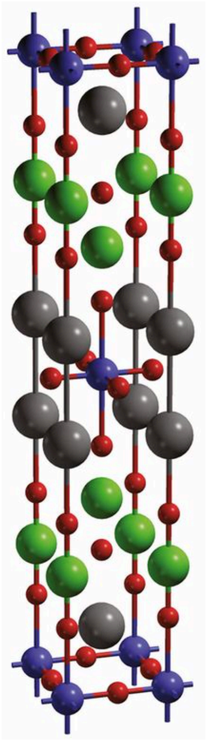
A classical computer cannot even represent the state efficiently

A quantum computer cannot produce a complete description of the state, but by performing measurements, it can answer questions that (apparently) a classical computer cannot

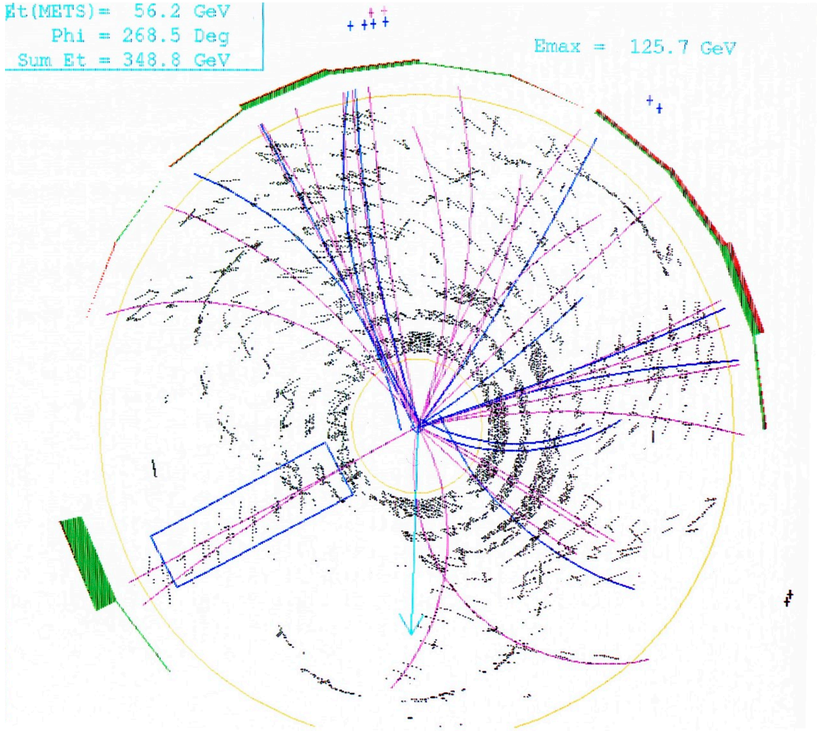
Computational quantum physics



quantum chemistry
(e.g., nitrogen fixation)

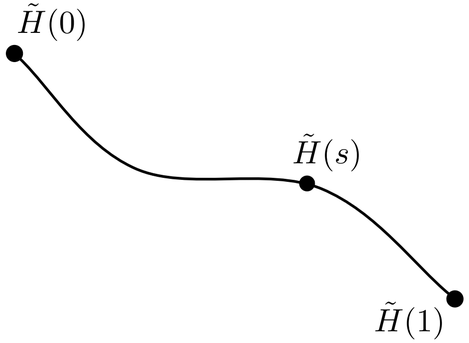


condensed matter physics/
properties of materials

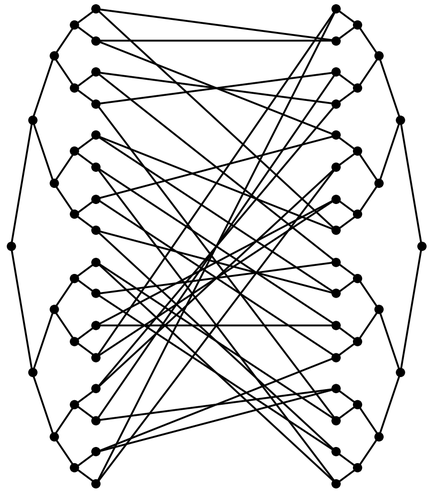


nuclear/particle
physics

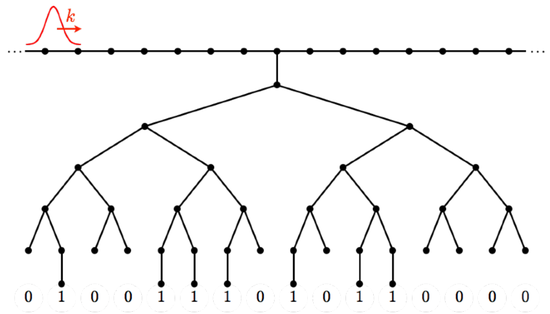
Implementing quantum algorithms



adiabatic optimization



exponential speedup by quantum walk



evaluating Boolean formulas

$$A|x\rangle = |b\rangle$$

linear/ differential equations, convex optimization

Quantum linear systems algorithm

Given an $N \times N$ system of linear equations $Ax = b$, find $x = A^{-1}b$

Quantum linear systems algorithm

Given an $N \times N$ system of linear equations $Ax = b$, find $x = A^{-1}b$

Classical (or quantum!) algorithms need time $\Omega(N)$ just to write down x

Quantum linear systems algorithm

Given an $N \times N$ system of linear equations $Ax = b$, find $x = A^{-1}b$

Classical (or quantum!) algorithms need time $\Omega(N)$ just to write down x

What if we change the model?

- A is sparse; given a black box that specifies the nonzero entries in any given row or column
- Can efficiently prepare a quantum state $|b\rangle$
- Goal is to prepare a state $|x\rangle \propto A^{-1}|b\rangle$

Quantum linear systems algorithm

Given an $N \times N$ system of linear equations $Ax = b$, find $x = A^{-1}b$

Classical (or quantum!) algorithms need time $\Omega(N)$ just to write down x

What if we change the model?

- A is sparse; given a black box that specifies the nonzero entries in any given row or column
- Can efficiently prepare a quantum state $|b\rangle$
- Goal is to prepare a state $|x\rangle \propto A^{-1}|b\rangle$

We can do this in time $\mathrm{poly}(\log N, 1/\epsilon, \kappa)$ where $\kappa = \|A\| \cdot \|A^{-1}\|$ [Harrow, Hassidim, Lloyd 09]

Quantum linear systems algorithm

Given an $N \times N$ system of linear equations $Ax = b$, find $x = A^{-1}b$

Classical (or quantum!) algorithms need time $\Omega(N)$ just to write down x

What if we change the model?

- A is sparse; given a black box that specifies the nonzero entries in any given row or column
- Can efficiently prepare a quantum state $|b\rangle$
- Goal is to prepare a state $|x\rangle \propto A^{-1}|b\rangle$

We can do this in time $\mathrm{poly}(\log N, 1/\epsilon, \kappa)$ where $\kappa = \|A\| \cdot \|A^{-1}\|$ [Harrow, Hassidim, Lloyd 09]

Algorithm estimates the eigenvalues of A (in superposition) and replaces them by their inverse (using postselection)

Quantum linear systems algorithm

Given an $N \times N$ system of linear equations $Ax = b$, find $x = A^{-1}b$

Classical (or quantum!) algorithms need time $\Omega(N)$ just to write down x

What if we change the model?

- A is sparse; given a black box that specifies the nonzero entries in any given row or column
- Can efficiently prepare a quantum state $|b\rangle$
- Goal is to prepare a state $|x\rangle \propto A^{-1}|b\rangle$

We can do this in time $\mathrm{poly}(\log N, 1/\epsilon, \kappa)$ where $\kappa = \|A\| \cdot \|A^{-1}\|$ [Harrow, Hassidim, Lloyd 09]

Algorithm estimates the eigenvalues of A (in superposition) and replaces them by their inverse (using postselection)

Subsequent improvements do the same with complexity $\kappa\,\mathrm{poly}(\log(1/\epsilon))$ using variable-time amplitude amplification and LCU [Ambainis 12; Childs, Kothari, Somma 17]

Quantum algorithms for differential equations

Given a linear ODE $\frac{\mathrm{d}x}{\mathrm{d}t} = Ax + b$ with an initial condition $x(0)$, determine $x(T)$ at some time T

Quantum algorithms for differential equations

Given a linear ODE $\frac{\mathrm{d}x}{\mathrm{d}t} = Ax + b$ with an initial condition $x(0)$, determine $x(T)$ at some time T

Quantum simulation is the special case with $A = -iH$ (anti-Hermitian), $b = 0$

Quantum algorithms for differential equations

Given a linear ODE $\frac{\mathrm{d}x}{\mathrm{d}t} = Ax + b$ with an initial condition $x(0)$, determine $x(T)$ at some time T

Quantum simulation is the special case with $A = -iH$ (anti-Hermitian), $b = 0$

But we can handle more general ODEs using the quantum linear systems algorithm [Berry 14]

Quantum algorithms for differential equations

Given a linear ODE $\frac{\mathrm{d}x}{\mathrm{d}t} = Ax + b$ with an initial condition $x(0)$, determine $x(T)$ at some time T

Quantum simulation is the special case with $A = -iH$ (anti-Hermitian), $b = 0$

But we can handle more general ODEs using the quantum linear systems algorithm [Berry 14]

Main idea:

- Approximate the dynamics by a system of linear equations (e.g., finite difference method)
- Bound the approximation error, the condition number of the system, and the success probability of the procedure in terms of properties of the ODE

Quantum algorithms for differential equations

Given a linear ODE $\frac{\mathrm{d}x}{\mathrm{d}t} = Ax + b$ with an initial condition $x(0)$, determine $x(T)$ at some time T

Quantum simulation is the special case with $A = -iH$ (anti-Hermitian), $b = 0$

But we can handle more general ODEs using the quantum linear systems algorithm [Berry 14]

Main idea:

- Approximate the dynamics by a system of linear equations (e.g., finite difference method)
- Bound the approximation error, the condition number of the system, and the success probability of the procedure in terms of properties of the ODE

This approach comes with the caveats of the QLSA: we need an implicit description of the system, and we only produce a quantum encoding of the solution

Quantum algorithms for differential equations

Given a linear ODE $\frac{\mathrm{d}x}{\mathrm{d}t} = Ax + b$ with an initial condition $x(0)$, determine $x(T)$ at some time T

Quantum simulation is the special case with $A = -iH$ (anti-Hermitian), $b = 0$

But we can handle more general ODEs using the quantum linear systems algorithm [Berry 14]

Main idea:

- Approximate the dynamics by a system of linear equations (e.g., finite difference method)
- Bound the approximation error, the condition number of the system, and the success probability of the procedure in terms of properties of the ODE

This approach comes with the caveats of the QLSA: we need an implicit description of the system, and we only produce a quantum encoding of the solution

The norm of the solution cannot decay exponentially (postselection is PP-hard)

Quantum algorithms for differential equations

Given a linear ODE $\frac{\mathrm{d}x}{\mathrm{d}t} = Ax + b$ with an initial condition $x(0)$, determine $x(T)$ at some time T

Quantum simulation is the special case with $A = -iH$ (anti-Hermitian), $b = 0$

But we can handle more general ODEs using the quantum linear systems algorithm [Berry 14]

Main idea:

- Approximate the dynamics by a system of linear equations (e.g., finite difference method)
- Bound the approximation error, the condition number of the system, and the success probability of the procedure in terms of properties of the ODE

This approach comes with the caveats of the QLSA: we need an implicit description of the system, and we only produce a quantum encoding of the solution

The norm of the solution cannot decay exponentially (postselection is PP-hard)

Later work gives improved/generalized algorithms (time-dependent coefficients, BVPs, PDEs)

Nonlinear dynamics

What about *nonlinear* differential equations?

Nonlinear dynamics

What about *nonlinear* differential equations?

Leyton, Osborne 08: Algorithm for N-dimensional nonlinear ODEs for time T with complexity $\mathrm{poly}(\log N), \exp(T)$

Nonlinear dynamics

What about *nonlinear* differential equations?

Leyton, Osborne 08: Algorithm for N-dimensional nonlinear ODEs for time T with complexity $\mathrm{poly}(\log N), \exp(T)$

Main idea: Use multiple copies of the solution to represent polynomial nonlinearities

Nonlinear dynamics

What about *nonlinear* differential equations?

Leyton, Osborne 08: Algorithm for N-dimensional nonlinear ODEs for time T with complexity $\mathrm{poly}(\log N), \exp(T)$

Main idea: Use multiple copies of the solution to represent polynomial nonlinearities

Problem: These copies are used up as we evolve. By the no-cloning theorem, need to maintain all copies throughout the algorithm. This leads to exponential overhead.

Nonlinear dynamics

What about *nonlinear* differential equations?

Leyton, Osborne 08: Algorithm for N-dimensional nonlinear ODEs for time T with complexity $\mathrm{poly}(\log N), \exp(T)$

Main idea: Use multiple copies of the solution to represent polynomial nonlinearities

Problem: These copies are used up as we evolve. By the no-cloning theorem, need to maintain all copies throughout the algorithm. This leads to exponential overhead.

Nonlinear dynamics are computationally powerful. In particular, nonlinear variants of quantum mechanics can quickly solve hard problems (e.g., unstructured search) [Abrams, Lloyd 98; Aaronson 05; Childs, Young 16].

Nonlinear dynamics

What about *nonlinear* differential equations?

Leyton, Osborne 08: Algorithm for N-dimensional nonlinear ODEs for time T with complexity $\mathrm{poly}(\log N), \exp(T)$

Main idea: Use multiple copies of the solution to represent polynomial nonlinearities

Problem: These copies are used up as we evolve. By the no-cloning theorem, need to maintain all copies throughout the algorithm. This leads to exponential overhead.

Nonlinear dynamics are computationally powerful. In particular, nonlinear variants of quantum mechanics can quickly solve hard problems (e.g., unstructured search) [Abrams, Lloyd 98; Aaronson 05; Childs, Young 16].

So maybe there is a fundamental obstacle?

Problem statement

Quantum quadratic ODE problem. Consider an ODE $\frac{\mathrm{d}u}{\mathrm{d}t} = F_2 u^{\otimes 2} + F_1 u + F_0(t)$ with $u(t), F_0(t) \in \mathbb{R}^n,\ F_1 \in \mathbb{R}^{n\times n},\ F_2 \in \mathbb{R}^{n\times n^2}$. Assume we are given an oracle to prepare a quantum state proportional to $u(0) = u_{\mathrm{in}}$, and sparse matrix oracles for $F_0(t), F_1, F_2$. Let the eigenvalues λ_j of F_1 satisfy $\mathrm{Re}(\lambda_n) \le \cdots \le \mathrm{Re}(\lambda_1) < 0$. Parametrize the problem in terms of

$$\mathrm{R} := \frac{1}{|\mathrm{Re}(\lambda_1)|}\left(\|u_{\mathrm{in}}\|\|F_2\| + \frac{\|F_0\|}{\|u_{\mathrm{in}}\|}\right)$$

and assume the values $\|u_{\mathrm{in}}\|, \|F_0(t)\|, \|F_1\|, \|F_2\|, \mathrm{Re}(\lambda_1), \max_t \|F_0(t)\|, \max_t \|F_0'(t)\|$ are known.

Goal: Produce a quantum state proportional to $u(T)$ for some given $T > 0$.

Problem statement

Quantum quadratic ODE problem. Consider an ODE $\dfrac{\mathrm{d}u}{\mathrm{d}t} = F_2 u^{\otimes 2} + F_1 u + F_0(t)$ with $u(t), F_0(t) \in \mathbb{R}^n,\ F_1 \in \mathbb{R}^{n\times n},\ F_2 \in \mathbb{R}^{n\times n^2}$. Assume we are given an oracle to prepare a quantum state proportional to $u(0) = u_{\mathrm{in}}$, and sparse matrix oracles for $F_0(t), F_1, F_2$. Let the eigenvalues λ_j of F_1 satisfy $\mathrm{Re}(\lambda_n) \le \cdots \le \mathrm{Re}(\lambda_1) < 0$. Parametrize the problem in terms of

$$\mathrm{R} := \frac{1}{|\mathrm{Re}(\lambda_1)|}\left(\|u_{\mathrm{in}}\|\|F_2\| + \frac{\|F_0\|}{\|u_{\mathrm{in}}\|}\right)$$

and assume the values $\|u_{\mathrm{in}}\|, \|F_0(t)\|, \|F_1\|, \|F_2\|, \mathrm{Re}(\lambda_1), \max_t \|F_0(t)\|, \max_t \|F_0'(t)\|$ are known.

Goal: Produce a quantum state proportional to $u(T)$ for some given $T > 0$.

R quantifies the strength of the nonlinearity and driving relative to dissipation.
Qualitatively similar to Reynolds number.

Main result I (algorithm)

Main result I (algorithm)

Theorem 1. For $\mathrm{R} < 1$, there is a quantum algorithm for the quantum quadratic ODE problem with query and gate complexity

$$\frac{sqT^2}{\epsilon}\mathrm{poly}\left(\log T, \log n, \log(1/\epsilon)\right)$$

where s is the sparsity of the input, $q := \|u_{\mathrm{in}}\|/\|u(T)\|$, and ϵ is the error in the approximation of $u(T)/\|u(T)\|$.

Main result I (algorithm)

Theorem 1. For $\mathrm{R} < 1$, there is a quantum algorithm for the quantum quadratic ODE problem with query and gate complexity

$$\frac{sqT^2}{\epsilon}\mathrm{poly}\left(\log T, \log n, \log(1/\epsilon)\right)$$

where s is the sparsity of the input, $q := \|u_{\mathrm{in}}\|/\|u(T)\|$, and ϵ is the error in the approximation of $u(T)/\|u(T)\|$.

Ingredients:

- Carleman linearization (novel convergence analysis)
- Forward Euler discretization
- State preparation procedure
- Condition number and success probability analysis

Carleman linearization

Approximate the nonlinear ODE by an infinite sequence of linear ODEs in $u, u^{\otimes 2}, u^{\otimes 3}, \ldots$

Carleman linearization

Approximate the nonlinear ODE by an infinite sequence of linear ODEs in $u, u^{\otimes 2}, u^{\otimes 3}, \ldots$

Example: $\dfrac{\mathrm{d}u}{\mathrm{d}t} = au^2 + bu + c$

Carleman linearization

Approximate the nonlinear ODE by an infinite sequence of linear ODEs in $u, u^{\otimes 2}, u^{\otimes 3}, \ldots$

Example: $\dfrac{\mathrm{d}u}{\mathrm{d}t} = au^2 + bu + c \qquad \dfrac{\mathrm{d}u^2}{\mathrm{d}t} = 2u\dfrac{\mathrm{d}u}{\mathrm{d}t} = 2au^3 + 2bu^2 + 2cu \qquad \cdots$

Carleman linearization

Approximate the nonlinear ODE by an infinite sequence of linear ODEs in $u, u^{\otimes 2}, u^{\otimes 3}, \ldots$

Example: $\dfrac{\mathrm{d}u}{\mathrm{d}t} = au^2 + bu + c \qquad \dfrac{\mathrm{d}u^2}{\mathrm{d}t} = 2u\dfrac{\mathrm{d}u}{\mathrm{d}t} = 2au^3 + 2bu^2 + 2cu \qquad \cdots$

In general: $$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} A_1^1 & A_2^1 & & \\ A_1^2 & A_2^2 & \ddots & \\ & \ddots & \ddots & A_N^{N-1} \\ & & A_{N-1}^N & A_N^N \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} + \begin{pmatrix} F_0(t) \\ 0 \\ \vdots \\ 0 \end{pmatrix} \qquad y_j \approx u^{\otimes j}$$

$$\begin{aligned} A_{j+1}^j &= F_2 \otimes I^{\otimes j-1} + I \otimes F_2 \otimes I^{\otimes j-2} + \cdots + I^{\otimes j-1} \otimes F_2 \\ A_j^j &= F_1 \otimes I^{\otimes j-1} + I \otimes F_1 \otimes I^{\otimes j-2} + \cdots + I^{\otimes j-1} \otimes F_1 \\ A_{j+1}^j &= F_0(t) \otimes I^{\otimes j-1} + I \otimes F_0(t) \otimes I^{\otimes j-2} + \cdots + I^{\otimes j-1} \otimes F_0(t) \end{aligned}$$

Carleman linearization

Approximate the nonlinear ODE by an infinite sequence of linear ODEs in $u, u^{\otimes 2}, u^{\otimes 3}, \ldots$

Example: $\dfrac{\mathrm{d}u}{\mathrm{d}t} = au^2 + bu + c \qquad \dfrac{\mathrm{d}u^2}{\mathrm{d}t} = 2u\dfrac{\mathrm{d}u}{\mathrm{d}t} = 2au^3 + 2bu^2 + 2cu \qquad \cdots$

In general: $$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} A_1^1 & A_2^1 & & \\ A_1^2 & A_2^2 & \ddots & \\ & \ddots & \ddots & A_N^{N-1} \\ & & A_{N-1}^N & A_N^N \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} + \begin{pmatrix} F_0(t) \\ 0 \\ \vdots \\ 0 \end{pmatrix} \qquad y_j \approx u^{\otimes j}$$

$$\begin{aligned} A_{j+1}^j &= F_2 \otimes I^{\otimes j-1} + I \otimes F_2 \otimes I^{\otimes j-2} + \cdots + I^{\otimes j-1} \otimes F_2 \\ A_j^j &= F_1 \otimes I^{\otimes j-1} + I \otimes F_1 \otimes I^{\otimes j-2} + \cdots + I^{\otimes j-1} \otimes F_1 \\ A_{j+1}^j &= F_0(t) \otimes I^{\otimes j-1} + I \otimes F_0(t) \otimes I^{\otimes j-2} + \cdots + I^{\otimes j-1} \otimes F_0(t) \end{aligned}$$

Lemma. $N = O\big(\log(T\|F_2\|/\delta)/\log(1/\|u_{\mathrm{in}}\|)\big)$ suffices to approximate the solution within δ.

Main result 2 (lower bound)

Main result 2 (lower bound)

Previous results show hardness of simulating nonlinear quantum mechanics, but not for models with dissipation

Main result 2 (lower bound)

Previous results show hardness of simulating nonlinear quantum mechanics, but not for models with dissipation

Theorem 2. Assume $\mathrm{R} \geq \sqrt{2}$. Then there is an instance of the quantum quadratic ODE problem such that any quantum algorithm for producing a quantum state approximating $u(T)/\|u(T)\|$ with bounded error must have worst-case time complexity exponential in T.

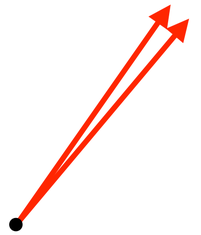
Main result 2 (lower bound)

Previous results show hardness of simulating nonlinear quantum mechanics, but not for models with dissipation

Theorem 2. Assume $\mathrm{R} \geq \sqrt{2}$. Then there is an instance of the quantum quadratic ODE problem such that any quantum algorithm for producing a quantum state approximating $u(T)/\|u(T)\|$ with bounded error must have worst-case time complexity exponential in T.

Ingredients:

- Hardness of distinguishing nonorthogonal quantum states



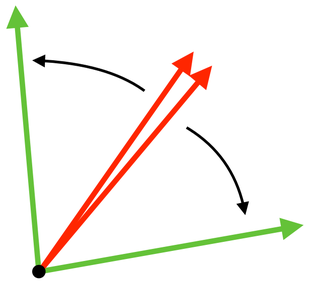
Main result 2 (lower bound)

Previous results show hardness of simulating nonlinear quantum mechanics, but not for models with dissipation

Theorem 2. Assume $\mathrm{R} \geq \sqrt{2}$. Then there is an instance of the quantum quadratic ODE problem such that any quantum algorithm for producing a quantum state approximating $u(T)/\|u(T)\|$ with bounded error must have worst-case time complexity exponential in T.

Ingredients:

- Hardness of distinguishing nonorthogonal quantum states
- Quadratic ODE that rapidly distinguishes nonorthogonal states



Hardness of state distinguishability

Lemma. Let $|\psi\rangle, |\phi\rangle$ be quantum states with $|\langle\psi|\phi\rangle| = 1 - \epsilon$. Suppose we are either given a black box that prepares $|\psi\rangle$ or a black box that prepares $|\phi\rangle$. Then any bounded-error protocol for determining whether the state is $|\psi\rangle$ or $|\phi\rangle$ must take time $\Omega(1/\epsilon)$.

Hardness of state distinguishability

Lemma. Let $|\psi\rangle, |\phi\rangle$ be quantum states with $|\langle\psi|\phi\rangle| = 1 - \epsilon$. Suppose we are either given a black box that prepares $|\psi\rangle$ or a black box that prepares $|\phi\rangle$. Then any bounded-error protocol for determining whether the state is $|\psi\rangle$ or $|\phi\rangle$ must take time $\Omega(1/\epsilon)$.

Proof:

- If we use the box k times, we produce states with overlap $(1 - \epsilon)^k$.
- These states have trace distance $\Theta(\sqrt{k\epsilon})$.
- By the Helstrom bound, need $k = \Omega(1/\epsilon)$ to distinguish with bounded error.

Distinguishing states with dissipative nonlinear dynamics

Distinguishing states with dissipative nonlinear dynamics

Consider the ODE $\dfrac{\mathrm{d}u}{\mathrm{d}t} = -u + \mathrm{R}u^2$

Distinguishing states with dissipative nonlinear dynamics

Consider the ODE $\dfrac{\mathrm{d}u}{\mathrm{d}t} = -u + \mathrm{R}u^2$

Solution: $u(t) = \dfrac{1}{\mathrm{R} - e^t(\mathrm{R} - 1/u(0))}$

Distinguishing states with dissipative nonlinear dynamics

Consider the ODE $\frac{\mathrm{d}u}{\mathrm{d}t} = -u + \mathrm{R}u^2$ Solution: $u(t) = \frac{1}{\mathrm{R} - e^t(\mathrm{R} - 1/u(0))}$

Let this act on both basis states of a qubit

Distinguishing states with dissipative nonlinear dynamics

Consider the ODE $\frac{\mathrm{d}u}{\mathrm{d}t} = -u + \mathrm{R}u^2$ Solution: $u(t) = \frac{1}{\mathrm{R} - e^t(\mathrm{R} - 1/u(0))}$

Let this act on both basis states of a qubit

The uniform superposition $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ does not evolve (up to normalization)

Distinguishing states with dissipative nonlinear dynamics

Consider the ODE $\frac{\mathrm{d}u}{\mathrm{d}t} = -u + \mathrm{R}u^2$ Solution: $u(t) = \frac{1}{\mathrm{R} - e^t(\mathrm{R} - 1/u(0))}$

Let this act on both basis states of a qubit

The uniform superposition $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ does not evolve (up to normalization)

But for small θ, the state $\cos(\frac{\pi}{4} + \theta)|0\rangle + \sin(\frac{\pi}{4} + \theta)|1\rangle$ changes exponentially in t

Distinguishing states with dissipative nonlinear dynamics

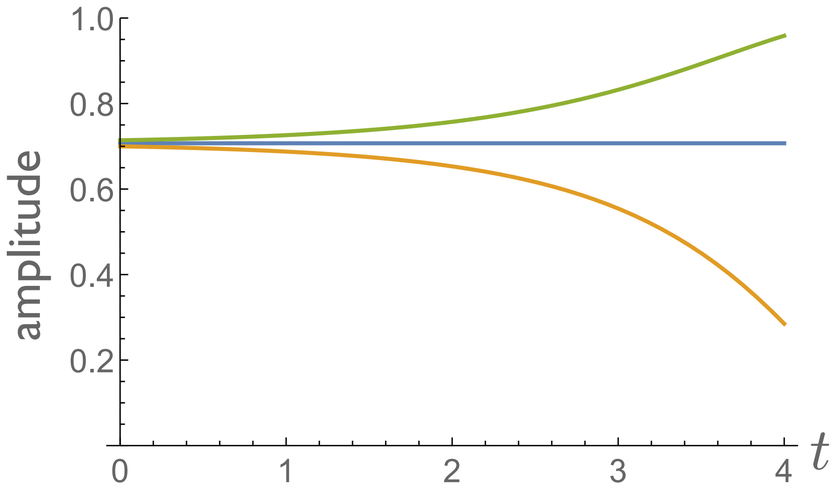
Consider the ODE $\dfrac{\mathrm{d}u}{\mathrm{d}t} = -u + \mathrm{R}u^2$ Solution: $u(t) = \dfrac{1}{\mathrm{R} - e^t(\mathrm{R} - 1/u(0))}$

Let this act on both basis states of a qubit

The uniform superposition $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ does not evolve (up to normalization)

But for small θ, the state $\cos(\frac{\pi}{4} + \theta)|0\rangle + \sin(\frac{\pi}{4} + \theta)|1\rangle$ changes exponentially in t

Example. $\mathrm{R} = \sqrt{2},\ \theta = 0.01$



Distinguishing states with dissipative nonlinear dynamics

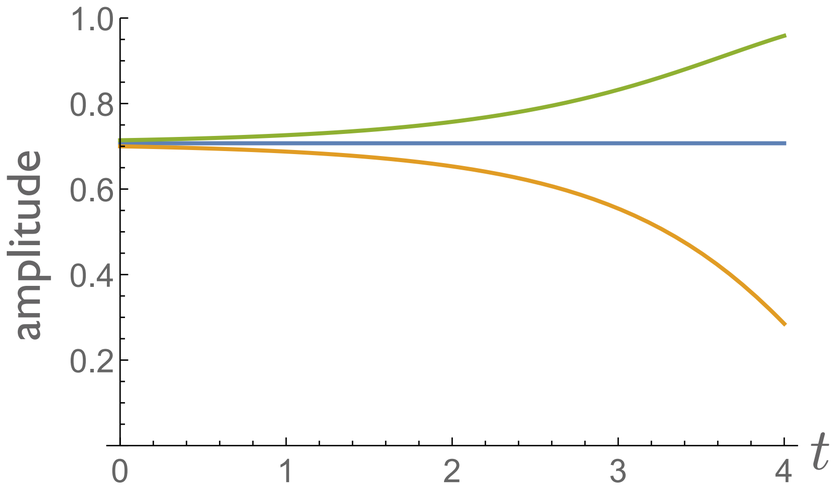
Consider the ODE $\frac{\mathrm{d}u}{\mathrm{d}t} = -u + \mathrm{R}u^2$ Solution: $u(t) = \frac{1}{\mathrm{R} - e^t(\mathrm{R} - 1/u(0))}$

Let this act on both basis states of a qubit

The uniform superposition $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ does not evolve (up to normalization)

But for small θ, the state $\cos(\frac{\pi}{4} + \theta)|0\rangle + \sin(\frac{\pi}{4} + \theta)|1\rangle$ changes exponentially in t

Example. $\mathrm{R} = \sqrt{2},\ \theta = 0.01$



In general, the time to separate states with overlap $1 - \epsilon$ by a constant amount is $O(\log(1/\epsilon))$.

Applications: Epidemiology

Applications: Epidemiology

SEIR model of a pandemic:

$$\frac{\mathrm{d}P_S}{\mathrm{d}t} = -\Lambda\frac{P_S}{P} - r_{\mathrm{vac}}P_S + \Lambda - r_{\mathrm{tra}}P_S\frac{P_I}{P}$$
$$\frac{\mathrm{d}P_E}{\mathrm{d}t} = -\Lambda\frac{P_E}{P} - \frac{P_E}{T_{\mathrm{lat}}} + r_{\mathrm{tra}}P_S\frac{P_I}{P}$$
$$\frac{\mathrm{d}P_I}{\mathrm{d}t} = -\Lambda\frac{P_I}{P} + \frac{P_E}{T_{\mathrm{lat}}} - \frac{P_I}{T_{\mathrm{inf}}}$$
$$\frac{\mathrm{d}P_R}{\mathrm{d}t} = -\Lambda\frac{P_R}{P} + r_{\mathrm{vac}}P_S + \frac{P_I}{T_{\mathrm{inf}}}$$

Applications: Epidemiology

SEIR model of a pandemic:

susceptible
$$\frac{\mathrm{d}P_S}{\mathrm{d}t} = -\Lambda\frac{P_S}{P} - r_{\mathrm{vac}}P_S + \Lambda - r_{\mathrm{tra}}P_S\frac{P_I}{P}$$

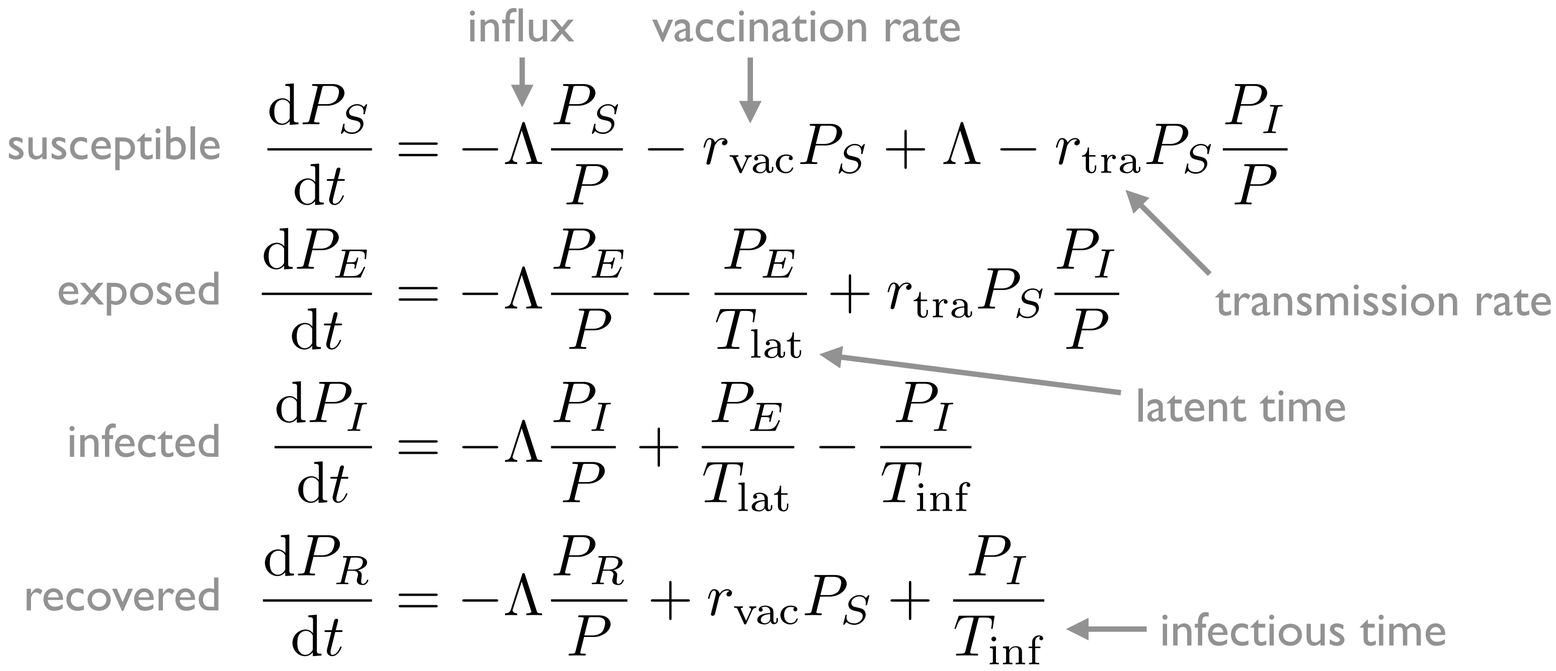
exposed
$$\frac{\mathrm{d}P_E}{\mathrm{d}t} = -\Lambda\frac{P_E}{P} - \frac{P_E}{T_{\mathrm{lat}}} + r_{\mathrm{tra}}P_S\frac{P_I}{P}$$

infected
$$\frac{\mathrm{d}P_I}{\mathrm{d}t} = -\Lambda\frac{P_I}{P} + \frac{P_E}{T_{\mathrm{lat}}} - \frac{P_I}{T_{\mathrm{inf}}}$$

recovered
$$\frac{\mathrm{d}P_R}{\mathrm{d}t} = -\Lambda\frac{P_R}{P} + r_{\mathrm{vac}}P_S + \frac{P_I}{T_{\mathrm{inf}}}$$

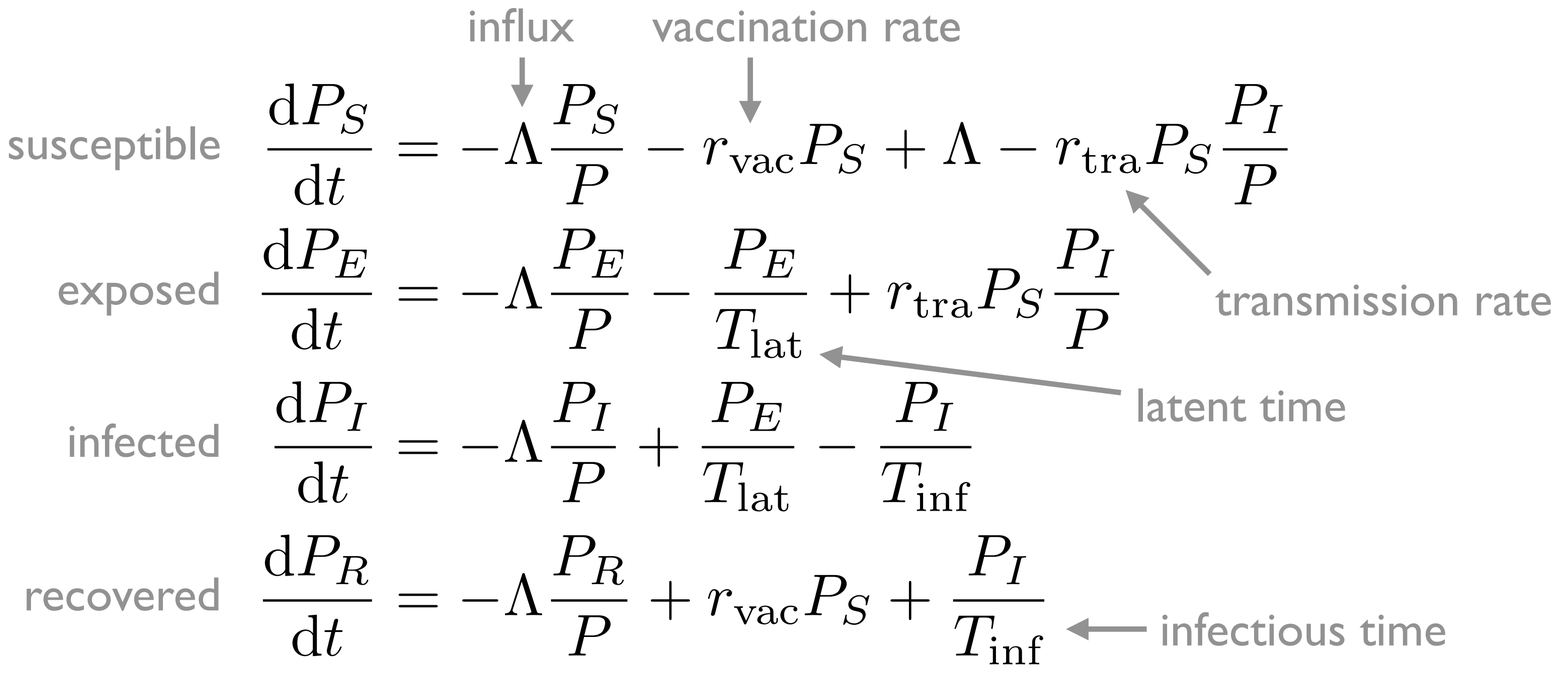
Applications: Epidemiology

SEIR model of a pandemic:



Applications: Epidemiology

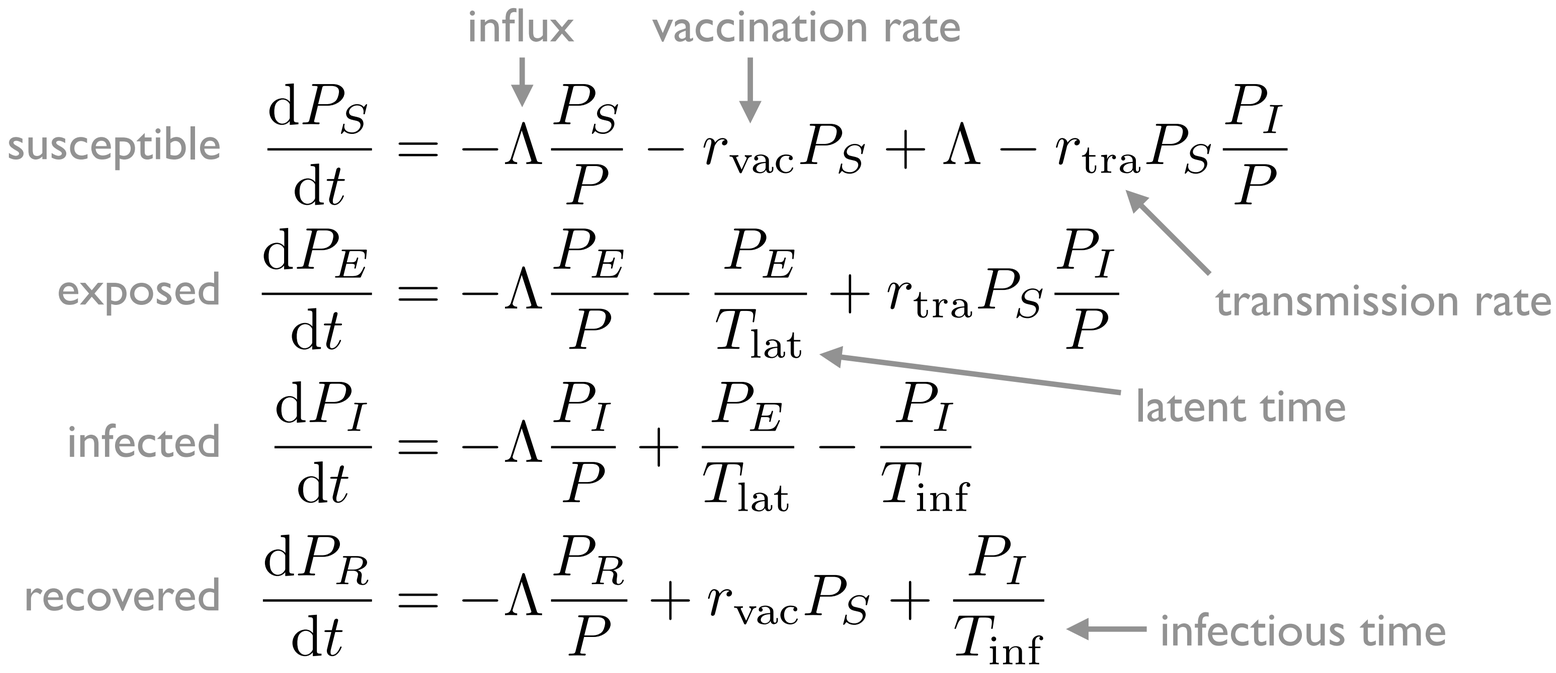
SEIR model of a pandemic:



Realistic parameters [Wang et al., JAMA 20] (with fairly rapid vaccination) can satisfy $\mathrm{R} < 1$

Applications: Epidemiology

SEIR model of a pandemic:



Realistic parameters [Wang et al., JAMA 20] (with fairly rapid vaccination) can satisfy $\mathrm{R} < 1$

A high-dimensional version can model many interacting cities

Applications: Fluid dynamics

Applications: Fluid dynamics

Forced viscous Burgers equation:

$$\partial_t u + u\partial_x u = \nu \partial_x^2 u + f$$

Applications: Fluid dynamics

Forced viscous Burgers equation:

$$\partial_t u + u\partial_x u = \nu \partial_x^2 u + f$$

Discretize space with central differences to give an ODE

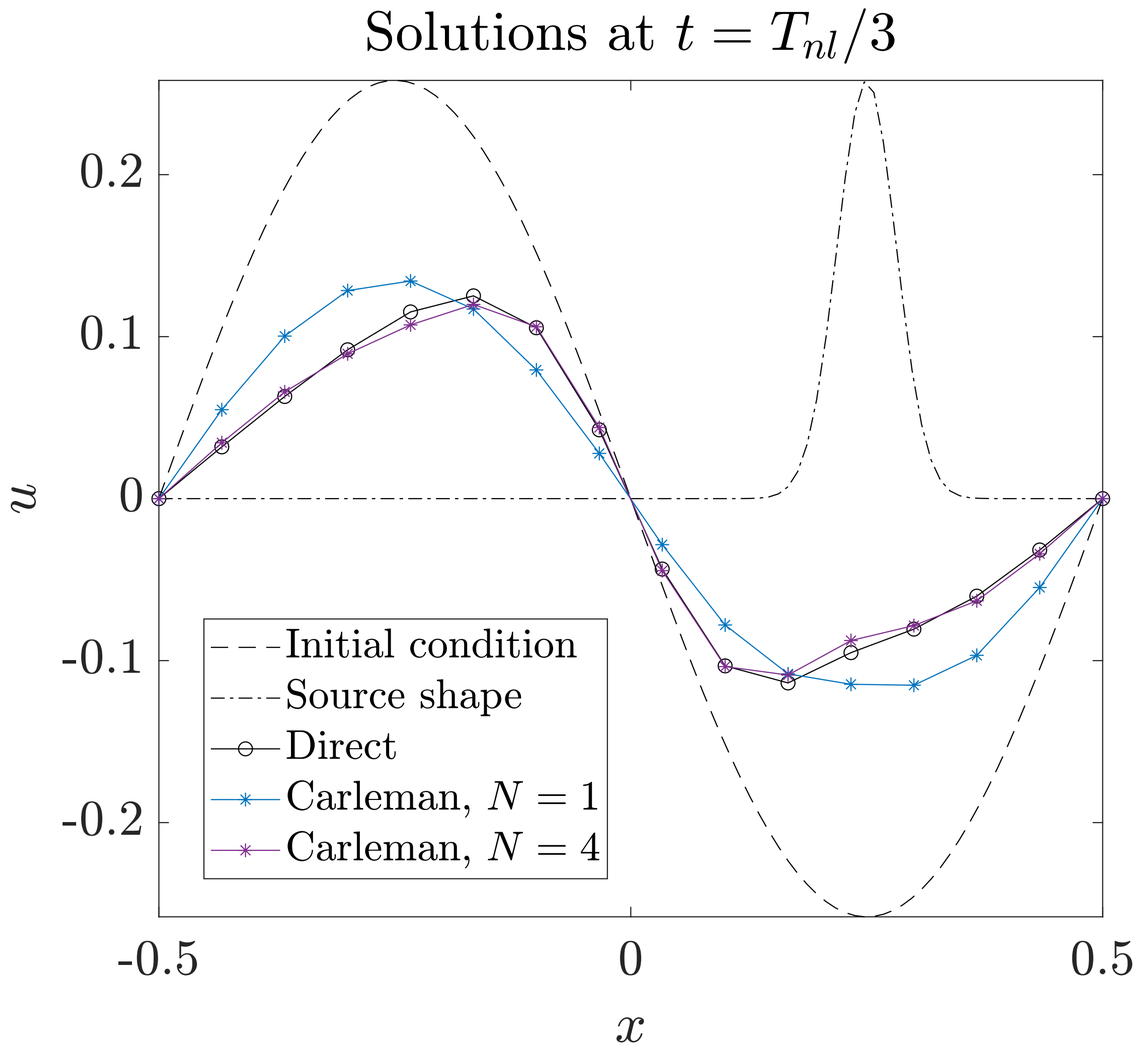
Applications: Fluid dynamics

Forced viscous Burgers equation:

$$\partial_t u + u\partial_x u = \nu \partial_x^2 u + f$$

Discretize space with central differences to give an ODE

Carleman method shows good convergence even for $\mathrm{R} \approx 40$



Summary

We have shown how to efficiently produce a quantum encoding of the solution of a system of dissipative nonlinear ODEs provided the nonlinearity and forcing are sufficiently weak relative to dissipation ($\mathrm{R} < 1$) and the solution does not decay exponentially.

Summary

We have shown how to efficiently produce a quantum encoding of the solution of a system of dissipative nonlinear ODEs provided the nonlinearity and forcing are sufficiently weak relative to dissipation ($\mathrm{R} < 1$) and the solution does not decay exponentially.

- Linear ODEs are BQP-complete, so this problem is classically hard

Summary

We have shown how to efficiently produce a quantum encoding of the solution of a system of dissipative nonlinear ODEs provided the nonlinearity and forcing are sufficiently weak relative to dissipation ($\mathrm{R} < 1$) and the solution does not decay exponentially.

- Linear ODEs are BQP-complete, so this problem is classically hard
- Exponential decay is an insurmountable obstacle (even for the linear case) due to the hardness of postselection

Summary

We have shown how to efficiently produce a quantum encoding of the solution of a system of dissipative nonlinear ODEs provided the nonlinearity and forcing are sufficiently weak relative to dissipation ($\mathrm{R} < 1$) and the solution does not decay exponentially.

- Linear ODEs are BQP-complete, so this problem is classically hard
- Exponential decay is an insurmountable obstacle (even for the linear case) due to the hardness of postselection
- Dissipative, non-driven ODEs necessarily decay exponentially, so no algorithm for long-time evolution can be efficient

Summary

We have shown how to efficiently produce a quantum encoding of the solution of a system of dissipative nonlinear ODEs provided the nonlinearity and forcing are sufficiently weak relative to dissipation ($\mathrm{R} < 1$) and the solution does not decay exponentially.

- Linear ODEs are BQP-complete, so this problem is classically hard
- Exponential decay is an insurmountable obstacle (even for the linear case) due to the hardness of postselection
- Dissipative, non-driven ODEs necessarily decay exponentially, so no algorithm for long-time evolution can be efficient
- Driven ODEs need not decay exponentially even under dissipation

Summary

We have shown how to efficiently produce a quantum encoding of the solution of a system of dissipative nonlinear ODEs provided the nonlinearity and forcing are sufficiently weak relative to dissipation ($\mathrm{R} < 1$) and the solution does not decay exponentially.

- Linear ODEs are BQP-complete, so this problem is classically hard
- Exponential decay is an insurmountable obstacle (even for the linear case) due to the hardness of postselection
- Dissipative, non-driven ODEs necessarily decay exponentially, so no algorithm for long-time evolution can be efficient
- Driven ODEs need not decay exponentially even under dissipation

We have also shown that there can be no efficient quantum algorithm for general nonlinear ODEs with $\mathrm{R} \geq \sqrt{2}$.

Summary

We have shown how to efficiently produce a quantum encoding of the solution of a system of dissipative nonlinear ODEs provided the nonlinearity and forcing are sufficiently weak relative to dissipation ($\mathrm{R} < 1$) and the solution does not decay exponentially.

- Linear ODEs are BQP-complete, so this problem is classically hard
- Exponential decay is an insurmountable obstacle (even for the linear case) due to the hardness of postselection
- Dissipative, non-driven ODEs necessarily decay exponentially, so no algorithm for long-time evolution can be efficient
- Driven ODEs need not decay exponentially even under dissipation

We have also shown that there can be no efficient quantum algorithm for general nonlinear ODEs with $\mathrm{R} \geq \sqrt{2}$.

However, numerical evidence suggests that the algorithm may be efficient for certain cases even with R much larger.

Open questions

- What can we say about efficiency/hardness for $1 \leq \mathrm{R} < \sqrt{2}$?
- Our algorithm has complexity quadratic in T. Can this be improved?
- Our algorithm has complexity $\tilde{O}(1/\epsilon)$, whereas other quantum algorithms for simulation/ linear equations/ODEs have complexity $\mathrm{poly}(\log(1/\epsilon))$. Can this be improved?
- Can we identify conditions under which the algorithm is efficient for larger R?
- End-to-end applications