Quantum algorithms for differential equations

Opportunities and challenges

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

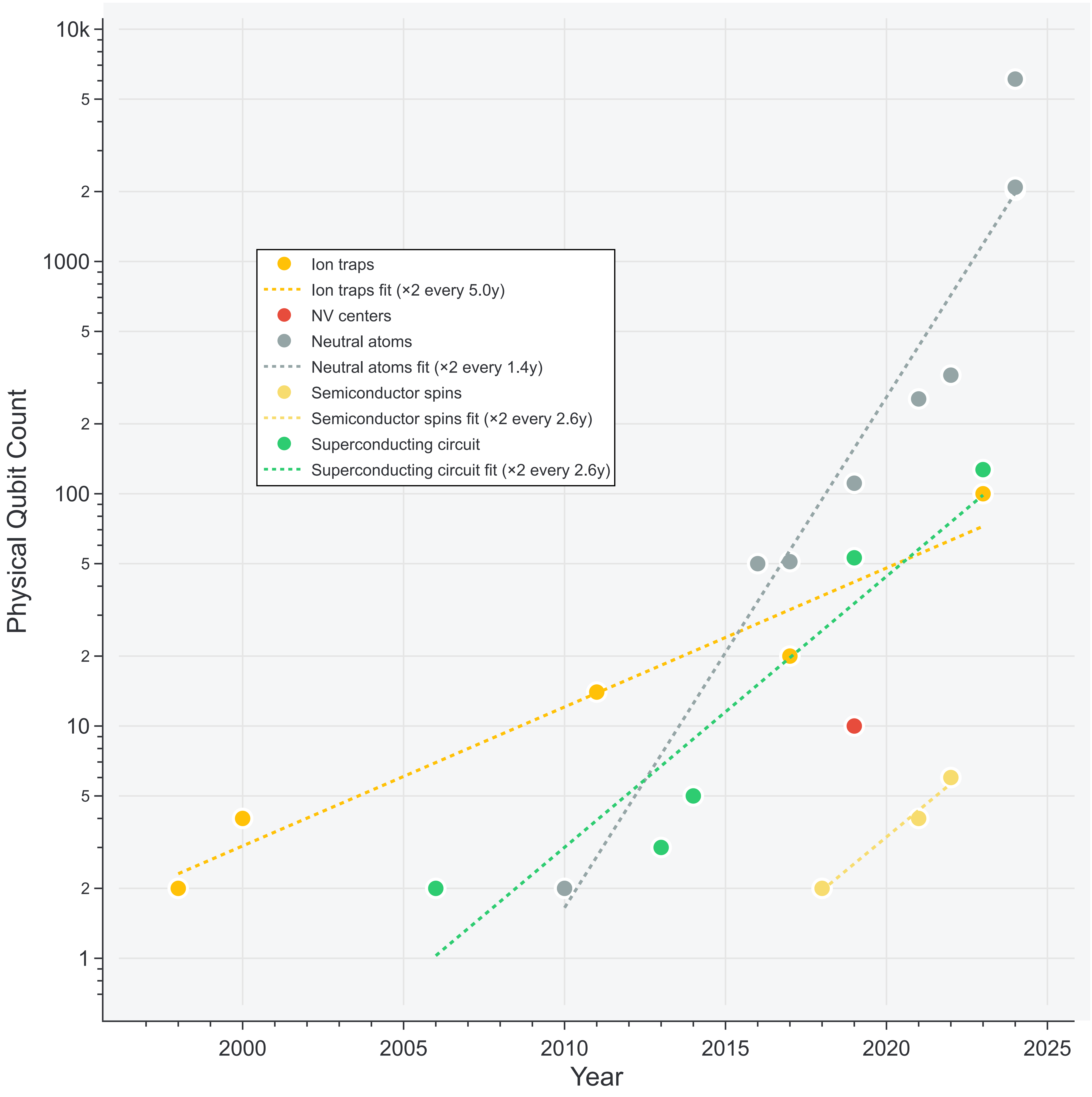
Institute for Robust
Quantum Simulation

Quantum computers

Store and process information in the states of quantum systems

Can solve certain problems dramatically faster than classical computers

Not yet capable of running useful algorithms, but advancing rapidly



[Le Regent, arXiv:2507.03678]

Quantum mechanics

States

$\psi \in \mathbb{C}^{2^n}$

amplitudes $\psi_x \in \mathbb{C}$
for $x \in \{0,1\}^n$

normalization:

$$\sum_{x \in \{0,1\}^n} |\psi_x|^2 = 1$$

Transformations

$\psi \mapsto U\psi$

unitary:
$U^\dagger U = I$

Measurements

$x \in \{0,1\}^n$
occurs with probability
$|\psi_x|^2$

Quantum cryptanalysis

Factoring integers is believed to be computationally difficult.

3107418240490043721350750035888567930037346022842727545720161948823206440518081504556346829671723286782437916272838033415471073108501919548529007337724822783525742386454014691736602477652346609

=

1634733645809253848443133883865090859841783670033092312181110852389333100104508151212118167511579

×

1900871281664822113126851573935413975471896789968515493666638539088027103802104498957191261465571

The security of modern communication relies on this assumption!

In 1994, Peter Shor showed that quantum computers can efficiently factor integers.

Main idea: Quantum computers can efficiently detect periodicity. The periodicity of the powers of a number modulo N is closely related to the prime factorization of N.

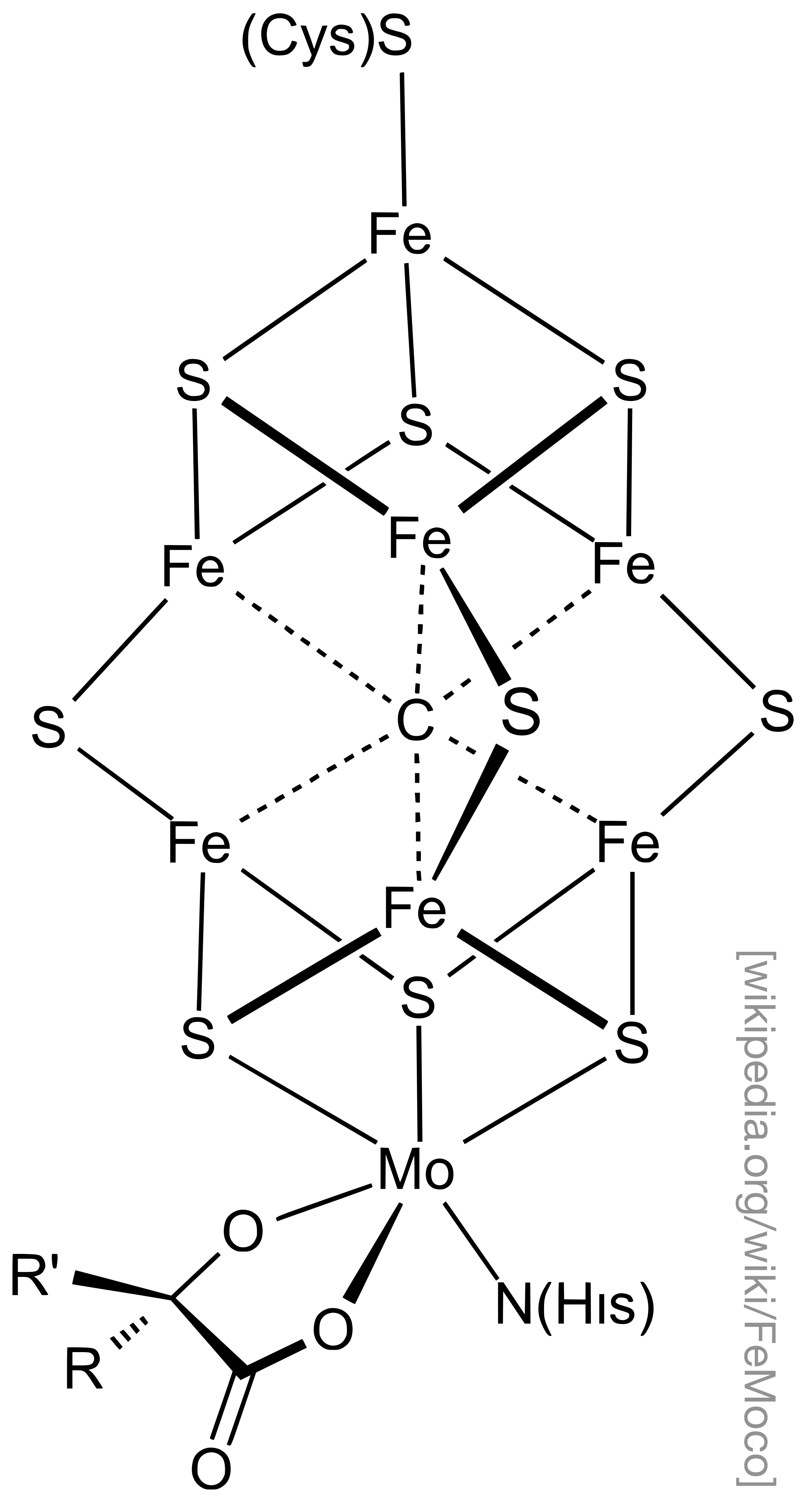
Quantum simulation

Quantum systems (e.g., in chemistry, materials science, nuclear/particle physics) are generically hard to simulate with classical computers.

The state space of n particles is exponential in n.

A quantum computer can efficiently represent and manipulate quantum states.

There are efficient quantum algorithms to simulate the dynamics of realistic quantum systems.



[wikipedia.org/wiki/FeMoco]

Other applications of quantum computers?

Computational number theory/algebra

Factoring integers, computing discrete logarithms, decomposing Abelian groups, approximating Gauss sums, shifted Legendre symbol problem, counting points on algebraic curves, Pell's equation, computing the unit group of an algebraic number field, …

Approximating topological invariants/partition functions

Jones polynomial, Turaev-Viro invariant, Potts model, …

Unstructured search and generalizations (polynomial speedup)

Collision finding, wide variety of graph/string/formula evaluation problems, …

Optimization? Unclear how much speedup is possible

High-dimensional linear algebra*

Surveys

Quantum algorithm zoo

Lecture notes on quantum algorithms

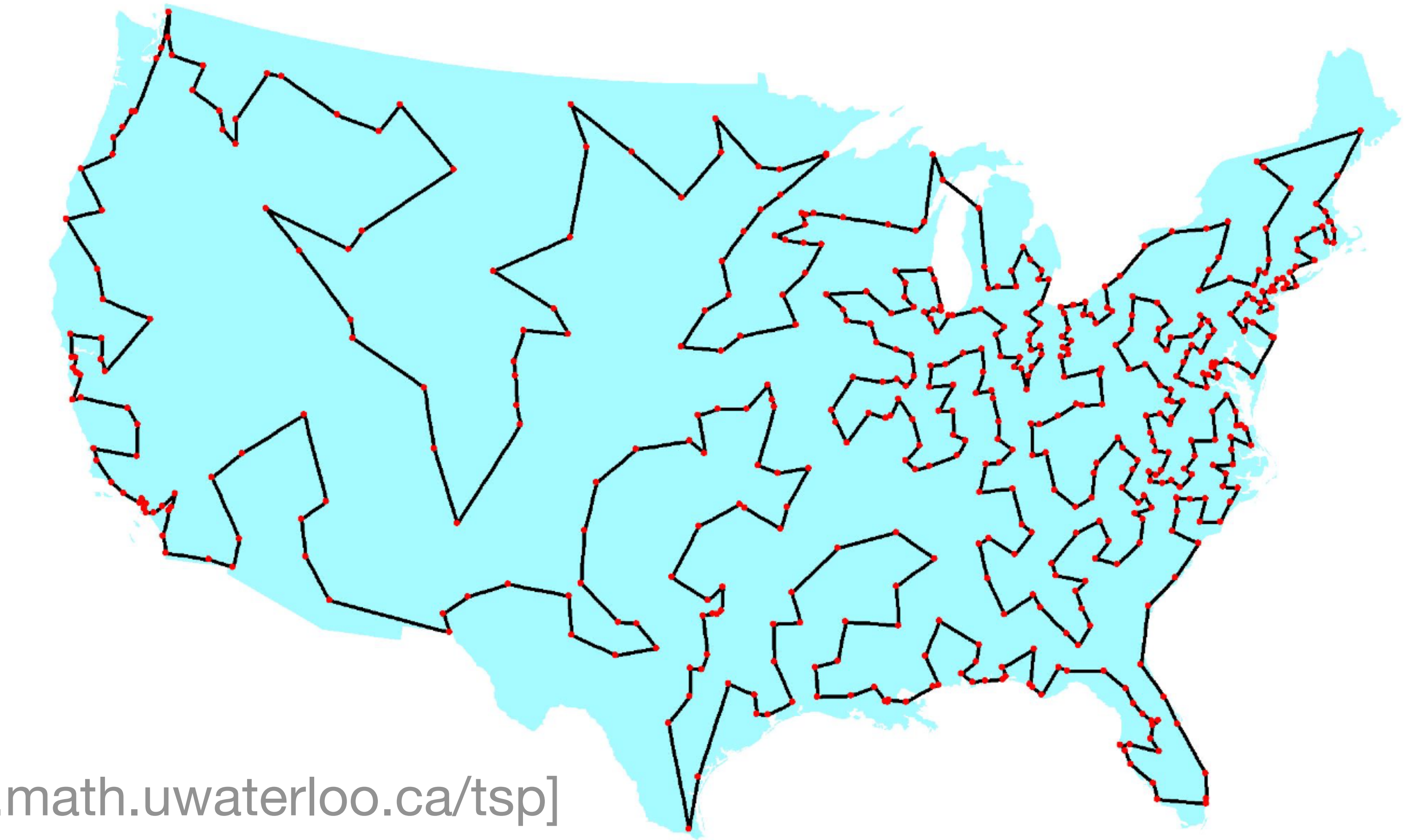
Dalzell et al.

What is a quantum computer *not* good for?

A quantum computer is not just a classical computer with a faster clock speed.

It can't simply try all possible solutions in parallel and find a valid one.

We do not expect efficient quantum algorithms for NP-hard problems.



[www.math.uwaterloo.ca/tsp]

Quantum computing for engineering

Can we develop quantum algorithms for computational problems in engineering?

Challenges:

- Quantum algorithms need to exploit specific structure.
- We are limited to theoretical analysis since we do not have large-scale quantum computers that can test algorithms empirically.
- It can be difficult to theoretically describe the performance of (quantum or classical) algorithms in practice.

One possible direction: quantum algorithms for differential equations

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 row/column sparse; given an efficient subroutine that specifies the nonzero entries in any row or column
  (more generally, given a *block encoding* of A, which describes it as a submatrix of a unitary transformation with an efficient quantum circuit)
- Can efficiently prepare a quantum state proportional to b
- Goal is to prepare a quantum state proportional to $A^{-1}b$

There is a quantum algorithm for this task running in time $\mathrm{poly}(\log N, 1/\epsilon, \kappa)$
where $\kappa := \|A\| \cdot \|A^{-1}\|$ [Harrow, Hassidim, Lloyd 09]

QLSA assumptions

Considering the ubiquity of linear systems in science, this should be widely applicable! But we need to satisfy the assumptions of the model.

Need implicit specifications of A and b (suitable for block encoding A and preparing b as a quantum state)

Explicit data describing A and b would have size $\mathrm{poly}(N)$ (exponential in the number of qubits) and could not be used to implement the algorithm efficiently

[wikipedia.org/wiki/Bridge]

Output of the algorithm is a quantum state. Need further processing for an end-to-end algorithm with classical output. What can we extract efficiently?

- Linear functional of the solution with polynomially small additive error
- But not the whole vector x, or even individual entries

QLSA under the hood

HHL algorithm:

- Extend A to a Hermitian operator $\begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix}$
- Apply quantum phase estimation to determine the eigenvalues of A (in superposition)
- Using postselection, replace the eigenvalue by its inverse
- Uncompute the eigenvalue

Complexity of this procedure is $O\big(\kappa^2 \operatorname{poly}(1/\epsilon, \log N)\big)$

Subsequent improvements achieve complexity $O(\kappa \operatorname{poly}(\log(1/\epsilon))$ using variable-time amplitude amplification [Ambainis 12] and linear combinations of unitaries [Childs, Kothari, Somma 17]. Later algorithms give further improvements using other ideas.

Quantum algorithms for differential equations

We can use the QLSA to give similar algorithms for differential equations

Example: Consider a system of linear ODEs $\frac{\mathrm{d}}{\mathrm{d}t}u(t) = Au(t) + b$, given a block encoding of A and the ability to prepare quantum states proportional to b and $u(0)$. Prepare a quantum state proportional to $u(T)$ for some desired final time T.

Applying a finite difference approximation gives a linear system. Under appropriate conditions, can use QLSA to produce an ϵ-approximation of the target state with complexity $\mathrm{poly}(\log N,\, T,\, 1/\epsilon)$. [Berry 14]

$$\begin{aligned} u(\delta) &= Au(0) + b \\ u(2\delta) &= Au(\delta) + b \\ &\vdots \\ u(T) &= Au(T-\delta) + b \end{aligned}$$

Generalizations give improved performance and handle more general systems (time-dependent coefficients, PDEs, boundary value problems, some nonlinear problems).

Same caveats as for the QLSA in general: implicit input, quantum output.

Limitations of quantum PDE algorithms

Montanaro and Pallister carried out a general comparison of classical and quantum algorithms for estimating a linear functional of the solution of an elliptic PDE boundary value problem in d dimensions, using the finite element method.

Quantum algorithms can use many mesh elements at low cost. But the number of mesh elements to achieve error ϵ for constant-dimensional problems typically scales as $\mathrm{poly}(1/\epsilon)$, and the cost of estimating an observable within ϵ is $\Omega(1/\epsilon)$, so the speedup is likely at most polynomial.

Performance depends on the effectiveness of preconditioning, which is problem-specific.

Algorithm	No preconditioning	Optimal preconditioning
Classical	$\widetilde{O}\left((\lvert u\rvert_2/\epsilon)^{(d+1)/2}\right)$	$\widetilde{O}\left((\lvert u\rvert_2/\epsilon)^{d/2}\right)$
Quantum	$\widetilde{O}\left(\lVert u\rVert \lvert u\rvert_2^2/\epsilon^3 + \lVert u\rVert_1 \lvert u\rvert_2/\epsilon^2\right)$	$\widetilde{O}\left(\lVert u\rVert_1/\epsilon\right)$

$\lVert u\rVert$	L^2 norm
$\lvert u\rvert_2 := (\sum_{\lvert\alpha\rvert=2} \lVert\partial^\alpha v\rVert^2)^{1/2}$	Sobolev 2-seminorm
$\lVert u\rVert_1 := \lvert u\rvert_0 + \lvert u\rvert_1$	Sobolev 1-norm

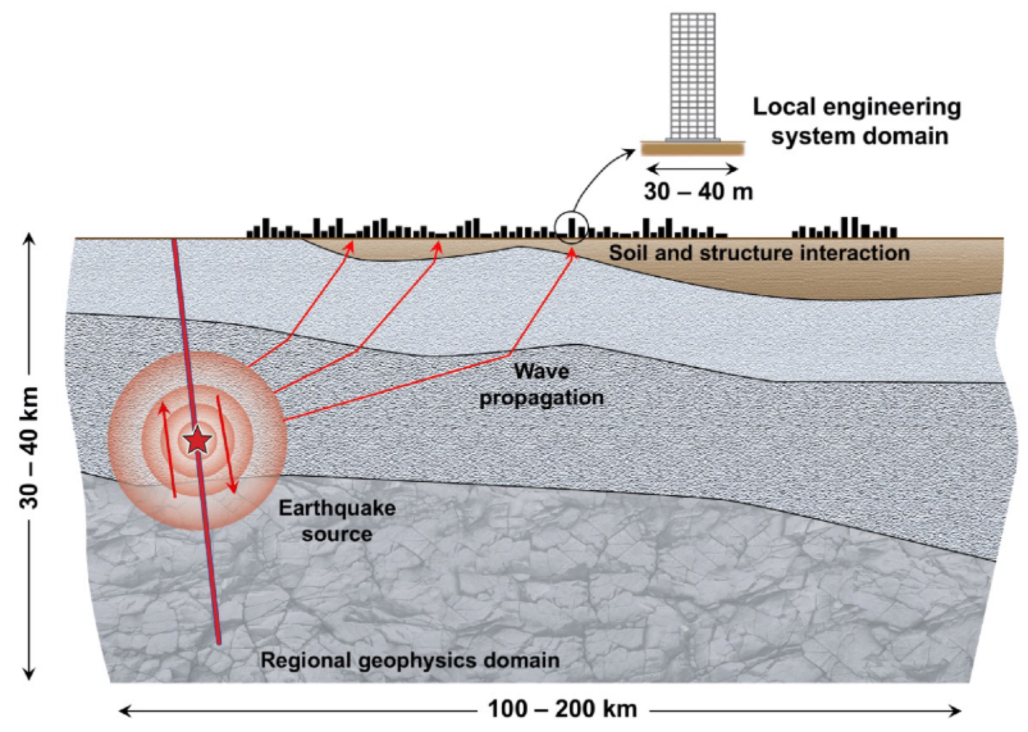
If the solution is smooth, expect at most a $d/2$ power speedup.

[Montanaro, Pallister 16]

Heterogeneous PDEs

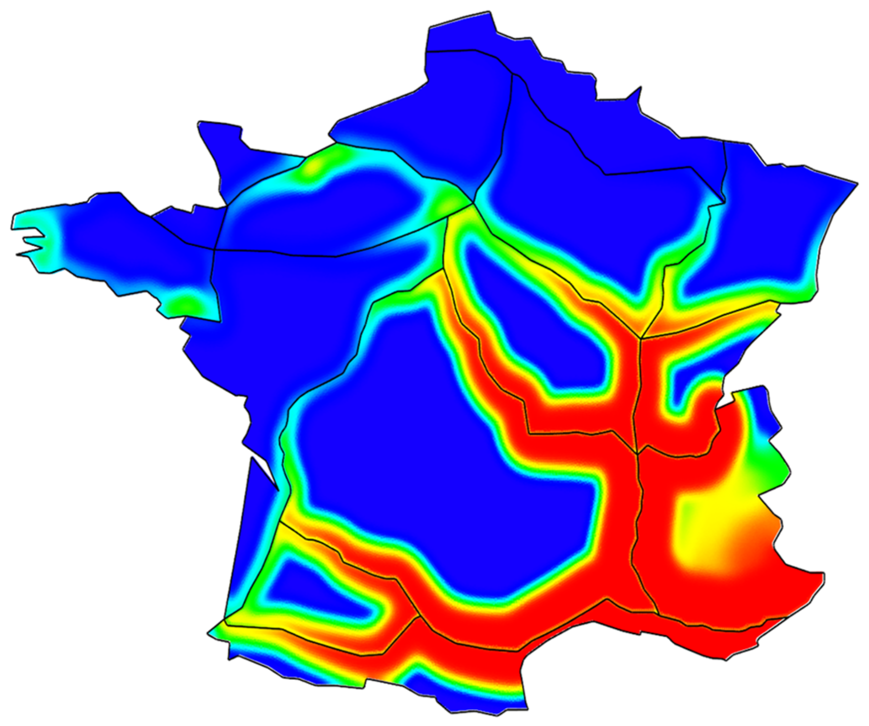
This suggests that to find larger quantum speedup in three spatial dimensions, we should look for problems with non-smooth solutions.

Consider problems involving heterogeneous media, with sharp material boundaries.



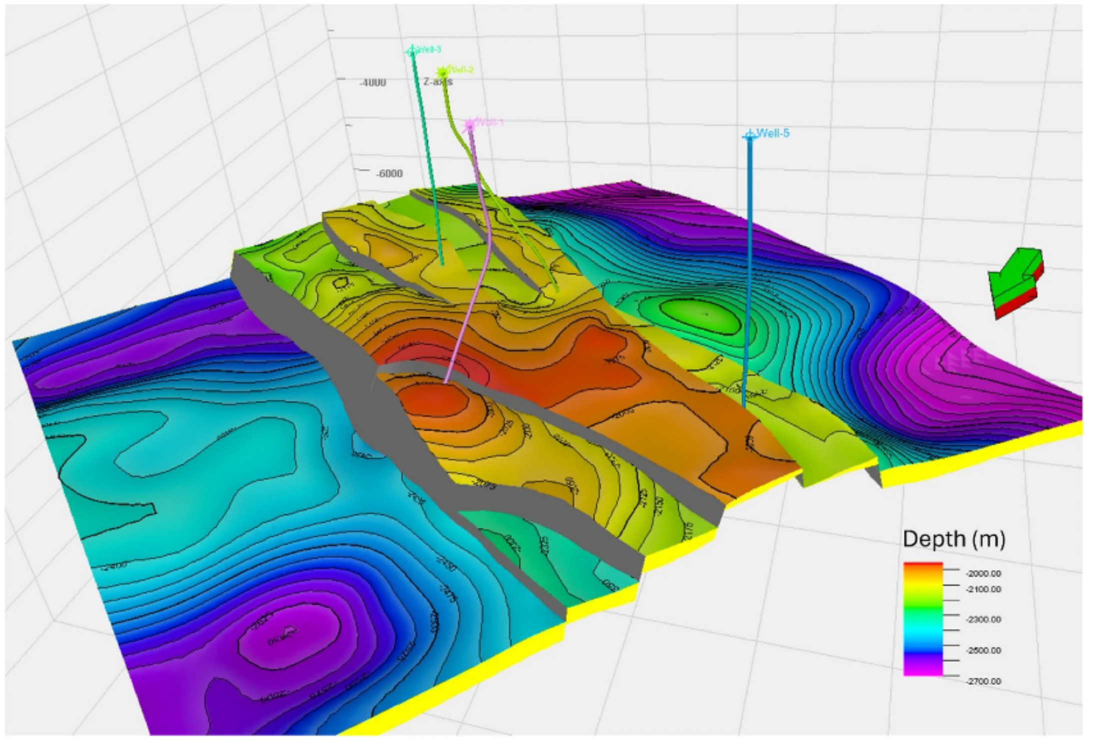
[exascaleproject.org]

wave propagation in geophysics



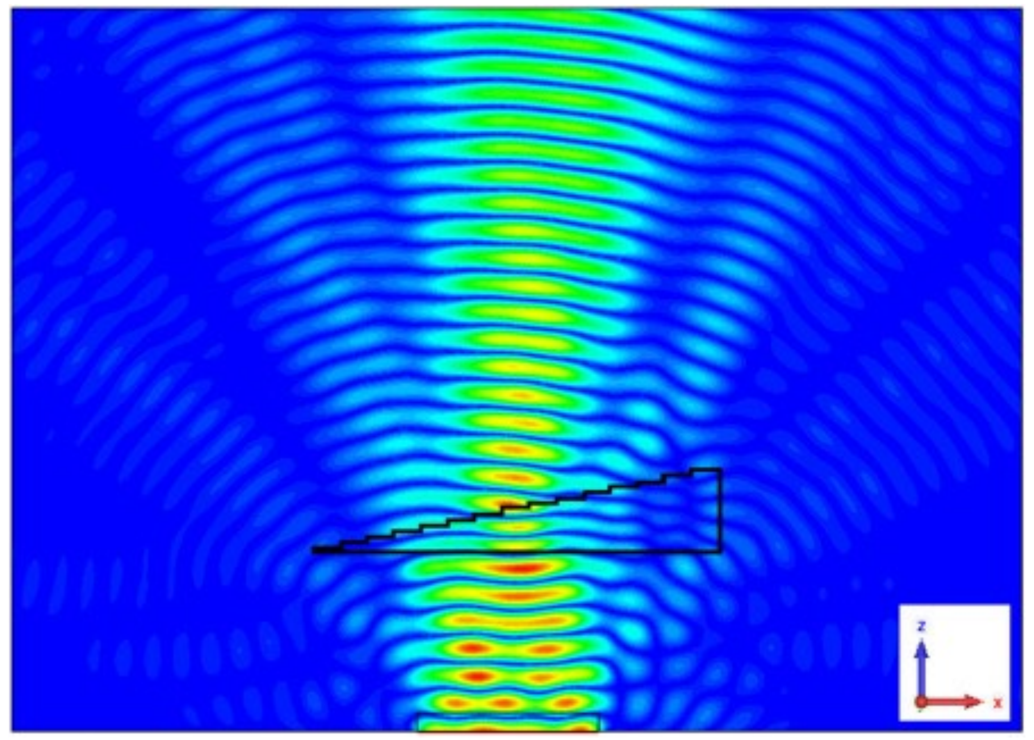
[doi.org/f8v5v8]

spatial distributions of populations



[doi.org/q9dm]

petroleum reservoir simulations



[doi.org/gs6d4q]

metamaterials

Case study: Neutron transport

Neutron diffusion eigenvalue problem: find the largest k satisfying

neutron flux

$$\Big(-\nabla \cdot (D(x)\nabla) + \Sigma_a(x)\Big)\phi(x) = \frac{1}{k}\nu\Sigma_f(x)\phi(x)$$

diffusion coefficient

absorption cross section

number of neutrons produced per fission event times fission cross section

Can be used to model the behavior of a nuclear reactor:

$k < 1$ subcritical

$k = 1$ critical

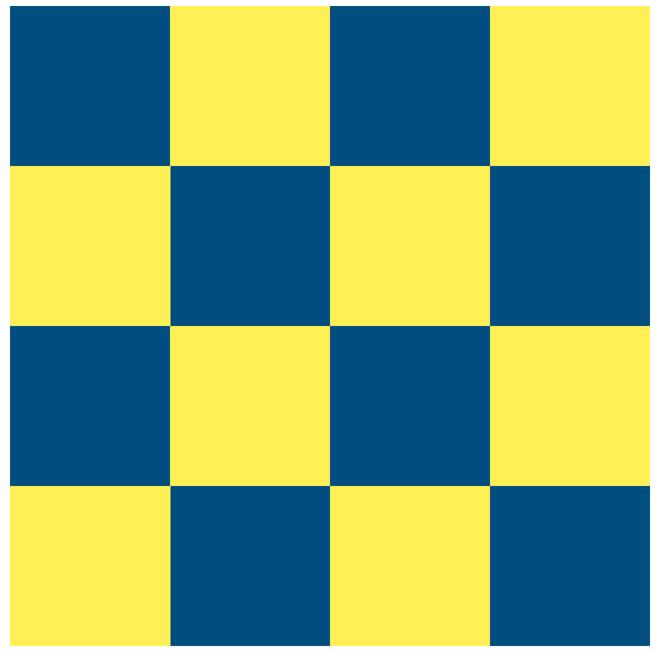
$k > 1$ supercritical

Determining k to high accuracy for a given geometry is a key problem in reactor design.

[nrc.gov]

Quantum speedup for neutron diffusion

Consider using the finite element method with a uniform mesh



Classical running time can be $O(1/\epsilon^p)$ for a high power p when the instance has sharp transitions between regions with different properties (numerics support this)

We develop a quantum algorithm with running time $O(\frac{1}{\epsilon}\,\mathrm{poly}(\frac{1}{\epsilon}))$

Caveats:

- Uniform FEM may not be the best classical algorithm
- High-precision calculations require more realistic models of neutron diffusion
- Even a large polynomial speedup may not be useful in practice

[Childs, Johnston, Kiedrowski, Vempati, Yu arXiv:2604.05098]

Structure of the quantum algorithm

- Solve the problem classically with coarse discretization
- Prepare a quantum state that interpolates this solution onto a finer mesh
- Determine the eigenvalue by applying quantum phase estimation to a block-encoded Hermitian operator that describes the system, constructed using
  - fast inversion [Tong, An, Wiebe, Lin 21] and
  - quantum preconditioning [Deiml, Peterseim 25]

[Childs, Johnston, Kiedrowski, Vempati, Yu arXiv:2604.05098]

Nonlinear differential equations

Leyton and Osborne [08] gave an algorithm for N-dimensional nonlinear ODEs acting 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 consumed as we evolve. By the quantum no-cloning theorem, need to maintain all copies throughout the algorithm. This leads to exponential overhead.

Many kinds of nonlinear dynamics are computationally powerful [Abrams, Lloyd 98; Aaronson 05; Childs, Young 16], so maybe this is a fundamental obstacle?

Quantum algorithm via Carleman linearization

Under similar assumptions as in previous algorithms for linear ODEs, we give a quantum algorithm for N-dimensional nonlinear ODEs with running time $\mathrm{poly}(T, \log N)$ provided $\mathrm{R} < 1$, where R is a parameter quantifying the strength of the nonlinearity and driving relative to dissipation.

Main idea:

- Approximate the nonlinear ODE in u by an infinite sequence of linear ODEs in $u, u^{\otimes 2}, u^{\otimes 3}, \ldots$ ("Carleman linearization")
- Bound the error from truncating to a finite number of copies
- Apply QLSA using forward Euler discretization, a state preparation procedure, and condition number/success probability analysis similar to the linear case

[Liu, Kolden, Krovi, Loureiro, Trivisa, Childs 21]

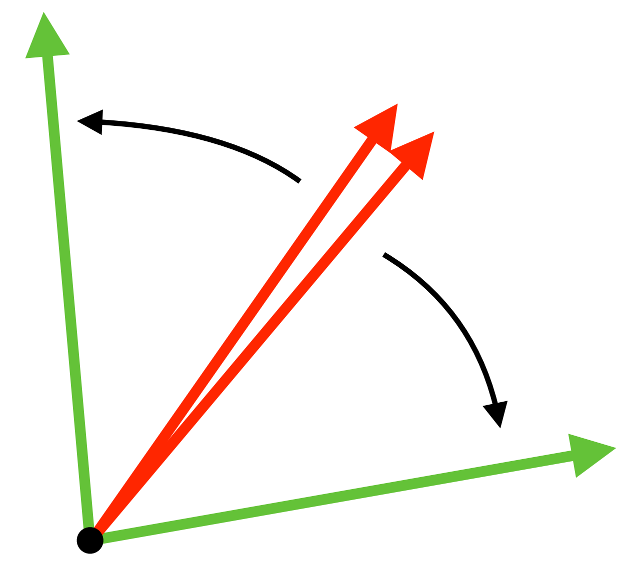
Limitations for nonlinear dynamics

This algorithm is only efficient if the dissipation is sufficiently strong.

Indeed, if $\mathrm{R} > 1$ then there is an instance of the problem such that any quantum algorithm for producing a quantum state proportional to $u(T)$ (up to small error) must have worst-case time complexity exponential in T.

Ingredients:

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



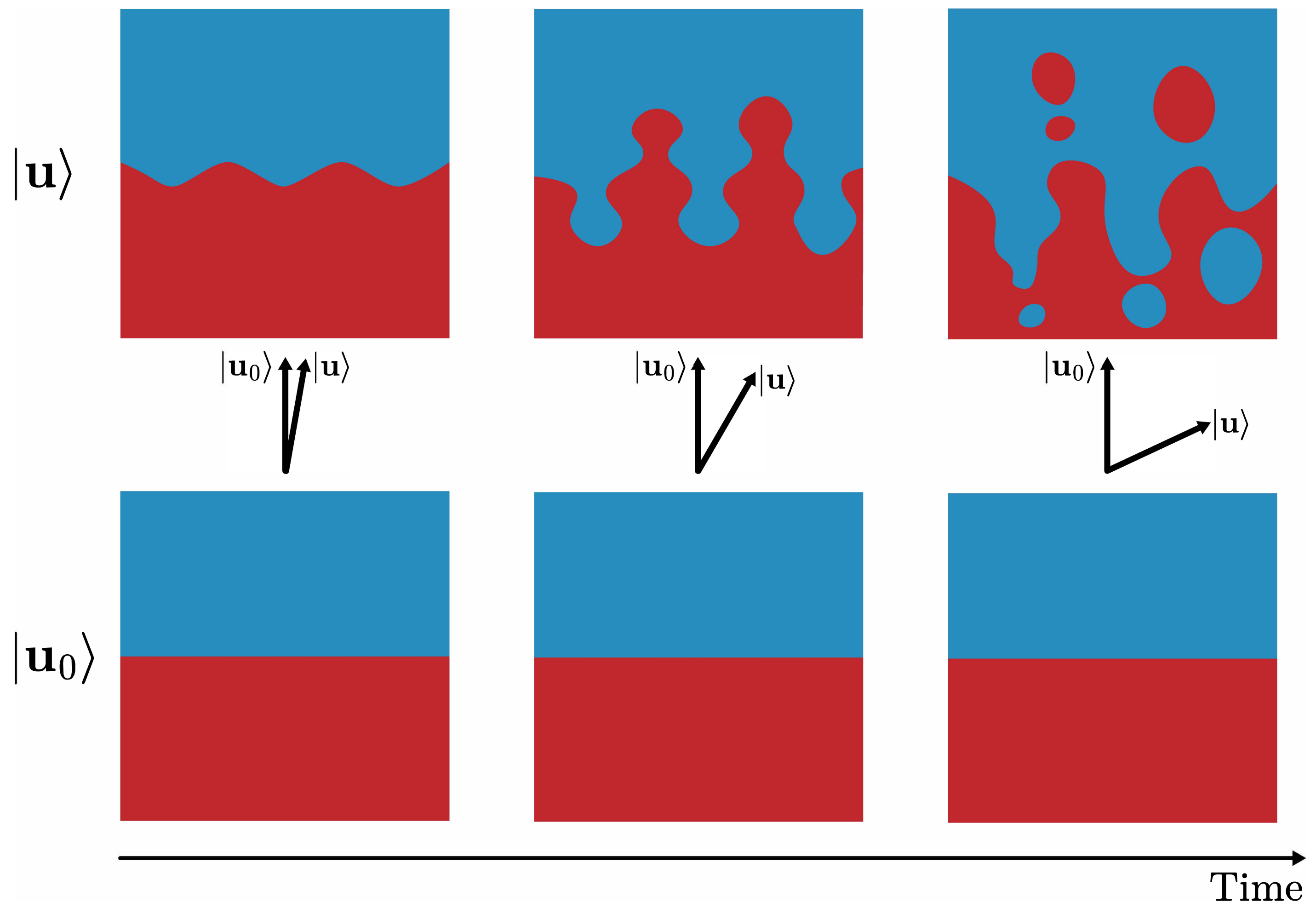
[Liu, Kolden, Krovi, Loureiro, Trivisa, Childs 21; Lewis, Eidenbenz, Nadiga, Subaşı 24]

Limitations for fluid dynamics

Efficient algorithms might still be possible for strongly nonlinear problems with suitable structure.

Several works have proposed quantum algorithms for fluid flows with high Reynolds number, and suggested that they might give exponential speedup.

But there are instances of the incompressible Euler equations with instabilities that provably map states with an initial overlap of $1 - \epsilon$ to states with constant overlap in time $\mathrm{poly}(\log(1/\epsilon))$, so again the complexity of simulating for time T must be exponential in T.



[Ameri, Carolan, Childs, Krovi arXiv:2603.12161]

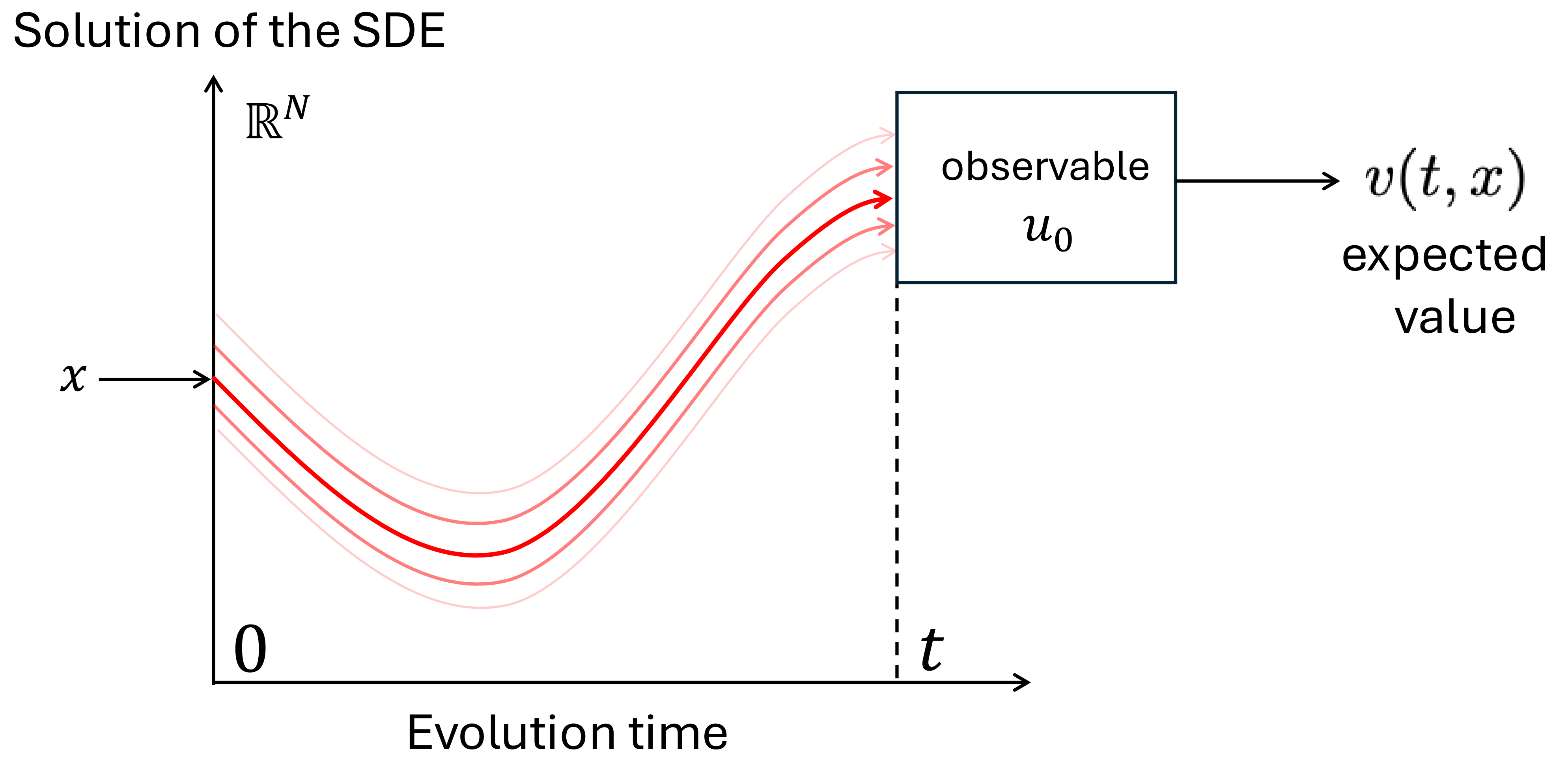
Quantum algorithm for noisy nonlinear dynamics

Nonlinear dynamics are hard to predict due to instabilities, but in practice we might care about properties that are robust to small perturbations.

Consider a dissipative nonlinear differential equation with a small amount of stochastic noise in the initial condition and dynamics.

Then there is an efficient quantum algorithm to estimate expectation values of the solution, given enough noise in the initial conditions (even in the strongly nonlinear regime where the Carleman linearization algorithm is inefficient).

Main idea: Use the Kolmogorov equation to describe the expected dynamics by a system of coupled quantum harmonic oscillators.



[Bravyi, Manson-Sawko, Zayats, Zhuk arXiv:2507.06198]

Summary

By representing high-dimensional vectors using quantum states, a quantum computer can produce a quantum encoding of the solution of a system of differential equations with exponentially less space, and sometimes exponentially faster, than a classical algorithm.

This savings comes at a cost: we need an implicit specification of the system and cannot produce an explicit solution.

There are quantum algorithms for handling nonlinear differential equations. Strong nonlinearities present challenges that are still being explored.

The potential for end-to-end advantage for practical problems remains unclear.

Outlook

There are many outstanding challenges to understand whether quantum computing can be a practical tool for engineering applications:

- Further develop quantum algorithms for differential equations (better preconditioning, linearization methods, optimized algorithms for particular kinds of equations, …).

- Better understand features that enable quantum speedup (high dimensions, non-smooth solutions, nonlinearity?, …).

- Develop end-to-end quantum algorithms for concrete applications and evaluate their performance compared to classical methods. This can benefit from more interaction between quantum algorithms specialists, experts on classical numerical methods, and application domain experts.

Large-scale quantum computers could significantly change how we explore the potential of these approaches, but the timeline for this is uncertain.