QUANTUM ALGORITHM FOR NONLINEAR ODES

Hari Krovi

Senior Scientist, Raytheon BBN Technologies

Advanced Modeling & Simulation (AMS) Seminar Series
NASA Ames Research Center, June 8th, 2021

Collaborators

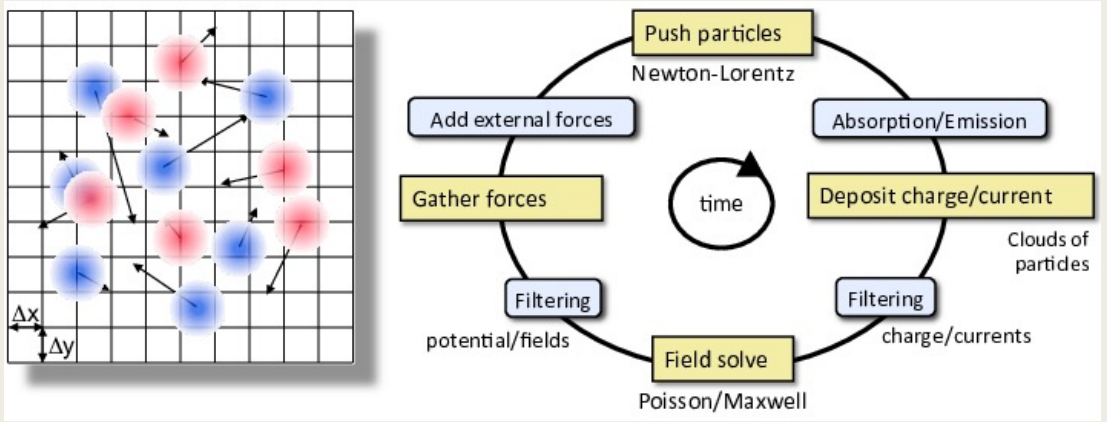
- Nuno Loureiro (MIT)
- Paola Cappellaro (MIT)
- Andrew Childs (UMD)
- Konstantina Trivisa (UMD)
- Jin-Peng Liu (student at UMD)
- Herman Kolden (Student at Norwegian University of Science and Technology)

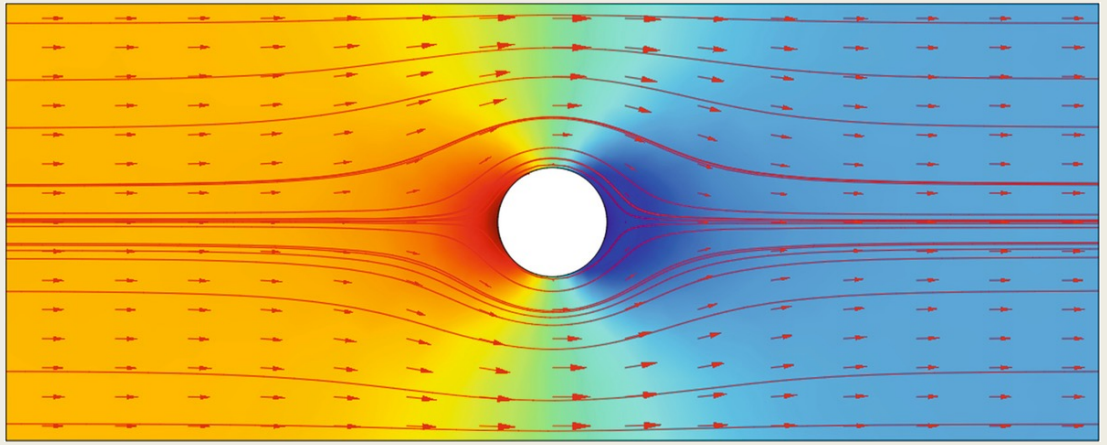





Main motivation

- Plasma simulations are very computationally intensive because they are very nonlinear.
- State-of-the-art classical algorithms such as particle-in-cell (PIC) take many processors and energy.
- There are many differential equations of interest to plasma physicists such as Vlasov equation, Navier-Stokes, viscous Burger's equation etc.
- These can be split into kinetic and fluid type equations because they either describe particles or fluid.
- The goal here is to design quantum algorithms (to be implemented on a fault-tolerant quantum computer) that can give speed-ups over classical algorithms.





Current classical techniques

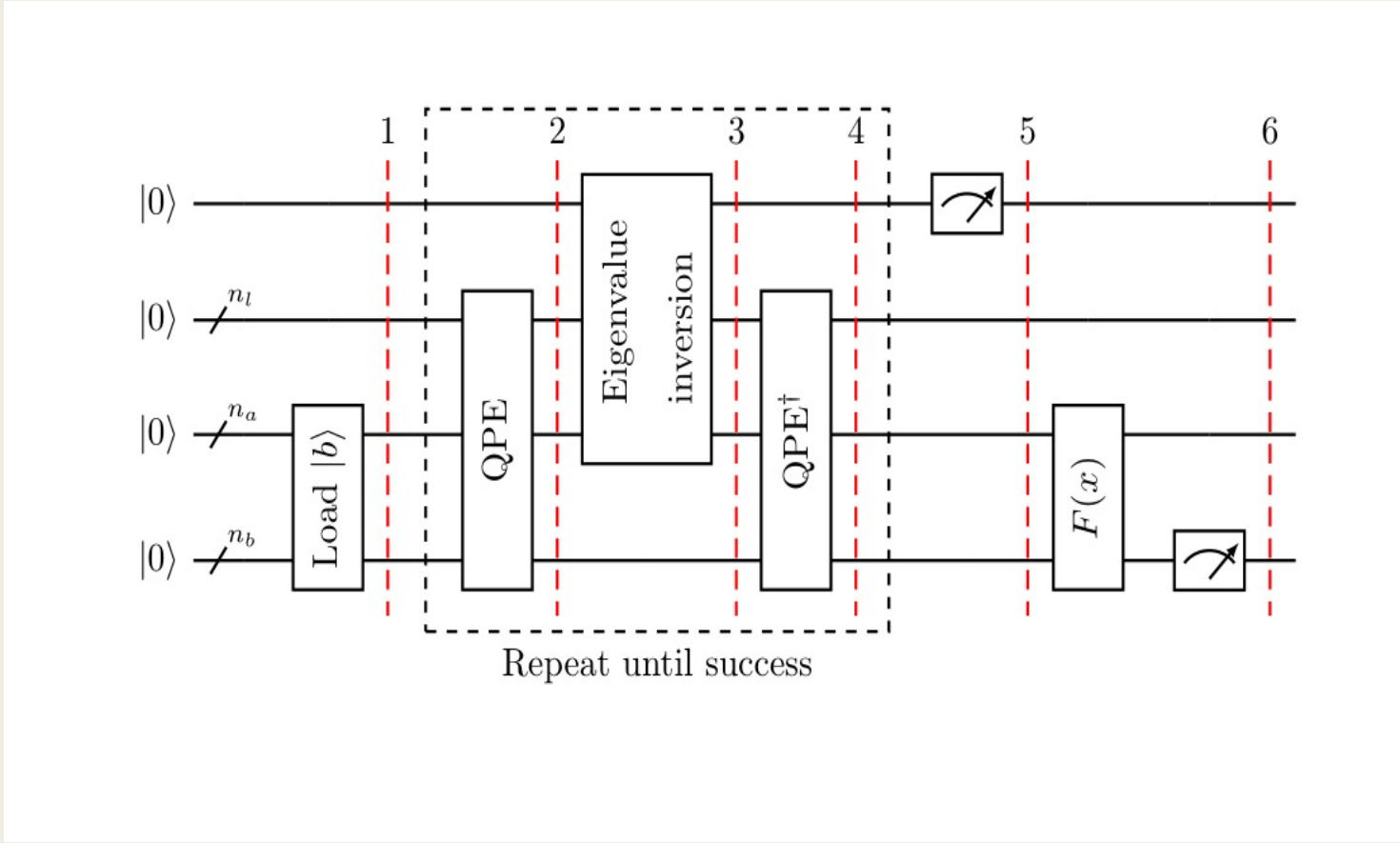
- Classical algorithms for plasma simulations are mainly divided into PIC type and transform based methods.
- PIC methods are very parallelizable since each cell can be treated on its own.
- There are usually many cells (such as a million) to keep track of, however.
- PIC methods currently use 300,000 processors (Dawson cluster) and 10MW of power (Fujitsu cluster).
- Transform methods are slower and not parallelizable, but give a higher accuracy.
- Fluid type models allow for the averaging of certain quantities so that only the important ones remain.

State of the art in quantum algorithms

- Quantum algorithms for differential equations are tackled by converting them to linear systems.
- If the conditions are right (such as well-behaved condition number), then one can get an exponential speed-up in the dimension and accuracy parameter.
- An important caveat is that the solution is in the form of a quantum state.
- Asking what the largest entries are or largest Fourier coefficient can be answered efficiently.
- Specific components of the solution vector are not efficient to produce.

Quantum linear systems algorithm

- Original algorithm is based on applying quantum phase estimation.
- This puts the eigenvalues (or phases) into the ancilla register.
- Using the ancilla register, compute the inverse transform.
- Then undo phase estimation.
- The solution vector remains as a quantum state.



Caveats and conditions for QLSA to solve Ax=b

- **Conditions:**
  - *Need to be able to prepare b as a quantum state.*
  - *Need to be able to efficiently implement the matrix A.*
  - *Condition number of A must be constant or polynomially increasing (not exponentially large).*
- **Caveats:**
  - *The solution is obtained as a quantum state and measuring to get every component will take exponential time.*
  - *Only useful if one wants information about statistics or averages.*

Improvements to QLSA

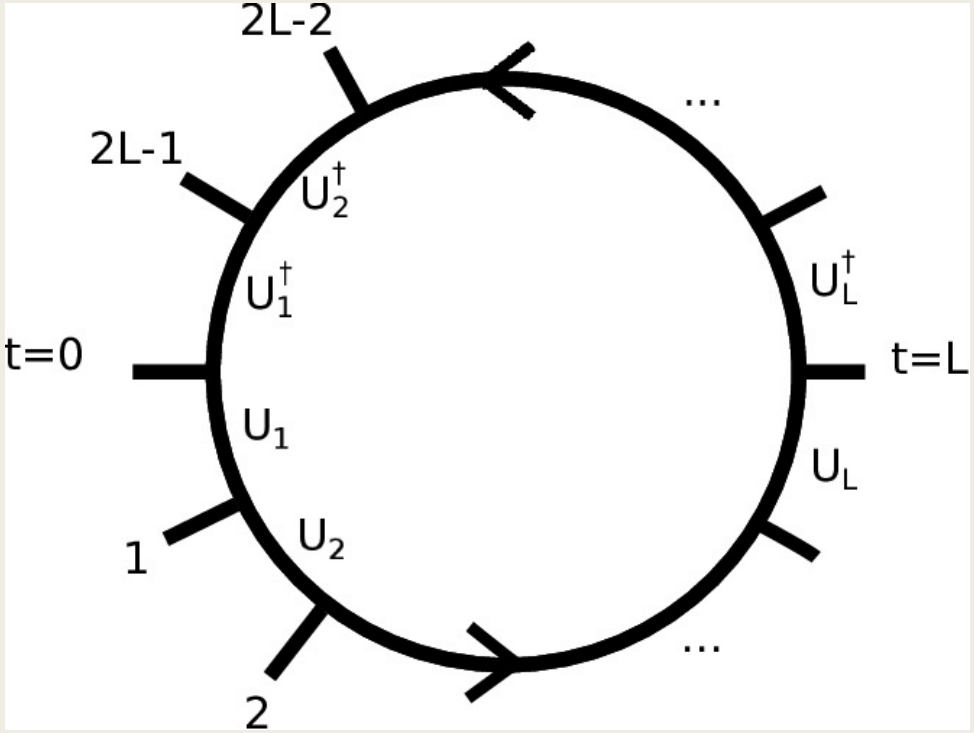
- There are several improvements to the original algorithm (HHL) in terms of parameters.
- The condition number dependence can be improved from κ^2 to $\kappa \log(\kappa)$.
- The dependence on the accuracy ϵ can be improved from poly$(1/\epsilon)$ to polylog$(1/\epsilon)$.
- The matrix A need not be Hermitian. One can make non-Hermitian matrices into a larger Hermitian matrix.
- If A is sparse or can be written as a linear combination of unitaries (LCU), then it can simulated efficiently for this algorithm.

Quantum algorithm for linear ODEs

- To solve linear ODEs of the form $\dot{x} = Ax + b$, we can convert them into a linear systems problem by discretizing time and choosing a method of stepping.
- Examples of stepping methods are Euler method (explicit or implicit), Runge-Kutta, Crank-Nicholson and multi-step procedures.
- Each comes with its own advantages and caveats.
- Most are useful in a certain regime of stability and stiffness.
- Using this we can get an exponential speed-up for these equations in dimension and accuracy.

Method using Feynman clock register

- To simulate the time evolution of the differential equation, we keep track of the time step in a separate register.
- This allows the condition number of the resulting linear system to depend on the norm of A rather than the condition number of A.
- Extending this to other types of stepping procedures such as implicit Euler, Crank-Nicholson etc. is non-trivial.
- There are extensions to incorporate time-dependent constant terms for inhomogeneous equations



Extending to nonlinear ODEs

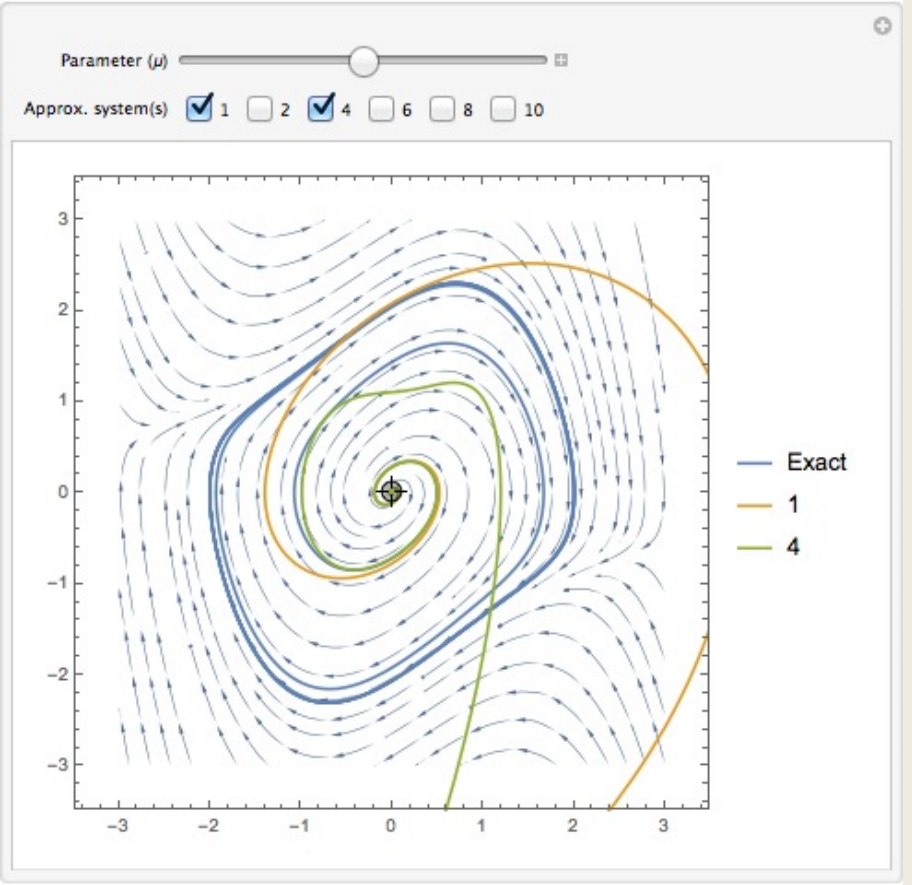
- Not much prior work exists on including nonlinearities mainly because it's hard to make everything unitary

$$\frac{du}{dt} = F_2 u^{\otimes 2} + F_1 u + F_0(t)$$

- Only rigorous work is Leyton-Osborne which uses post-selection to implement the nonlinearity.
- Unfortunately, this leads to an exponential time algorithm.
- It also needs an exponential number of copies of the initial quantum state.
- But it had the novelty of using the QLSA to solve nonlinear equations.
- We use another way to linearize the nonlinear equations using Carleman linearization.

Carleman linearization

- The general procedure to solve a nonlinear equation is the following.
  - *Linearize using some procedure (Leyton-Osborne use post-selection). In our work, we use Carleman linearization.*
  - *Then apply the quantum linear ODE solver to the resulting linear system.*
  - *Rigorously bound the condition number and run-time.*
- Carleman linearization introduces additional variables and constraints to linearize a nonlinear equation.
- Wolfram simulation of Van der Pol oscillator



$$\frac{d^2x}{dt^2} - \mu(1-x^2)\frac{dx}{dt} + x = 0$$

Amount of nonlinearity *R*

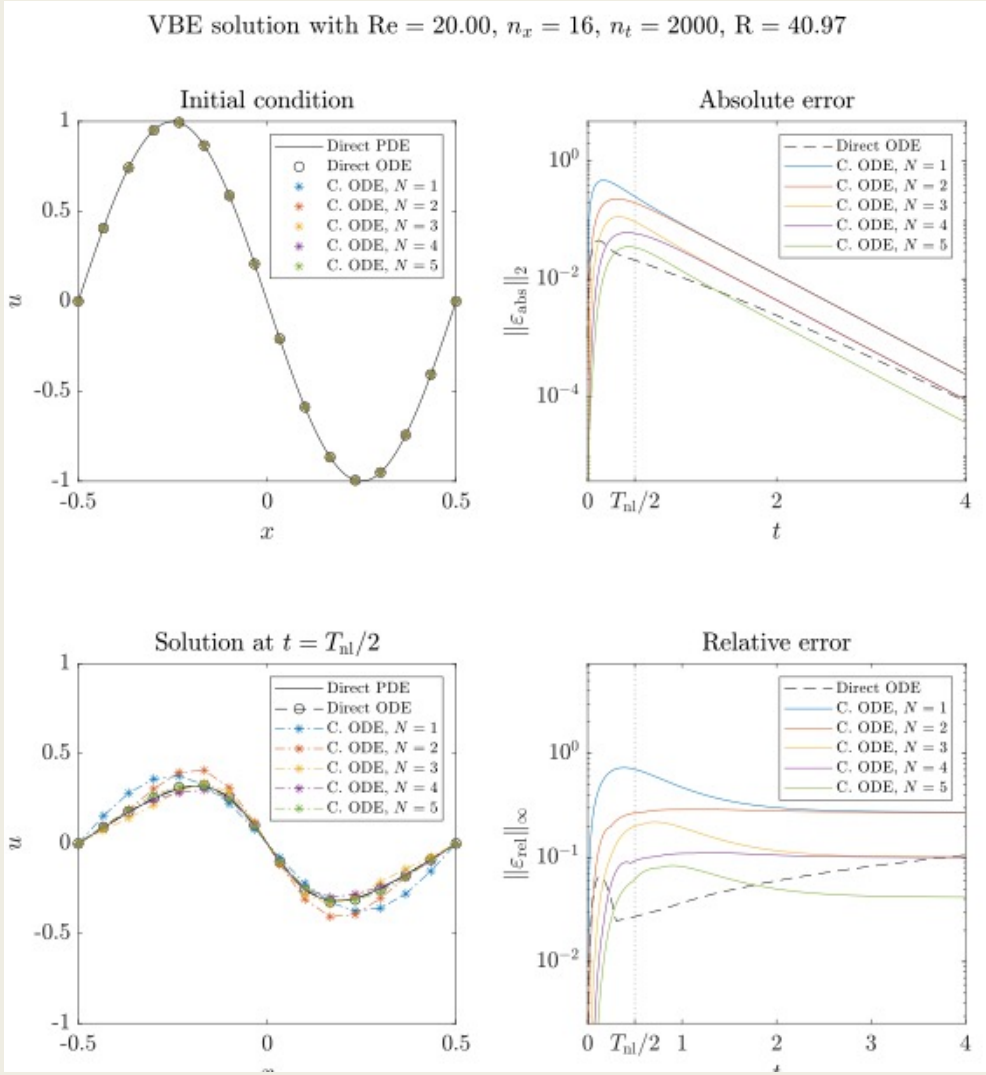
- It turns out the only for weakly nonlinear systems, there is a good convergence to the solution for Carleman linearization.
- To quantify this, we define *R* as follows.

$$\mathrm{R} := \frac{\|\mathbf{u}_{\mathrm{in}}\|\|F_2\|}{|\Re(\lambda_1)|}$$

- This is roughly the ratio of the norm of the nonlinear part to the norm of the linear part.
- When $R < 1$, we have convergence and an efficient algorithm.

Quantum algorithm for NLODEs

- We consider NLODEs with polynomial nonlinearities.
- Apply Carleman linearization to linearize it and determine the conditions of convergence.
- Then apply Euler method after determining the step size for convergence.
- Then apply QLSA to obtain the solution state.



Viscous Burgers equation

- We found that the convergence of the linearization procedure could be extended in some cases.
- Numerically, we found that for greater Reynolds numbers, Carleman convergence works.
- Explicitly proving this or finding other examples is important.
- R<1 is only a sufficient condition.

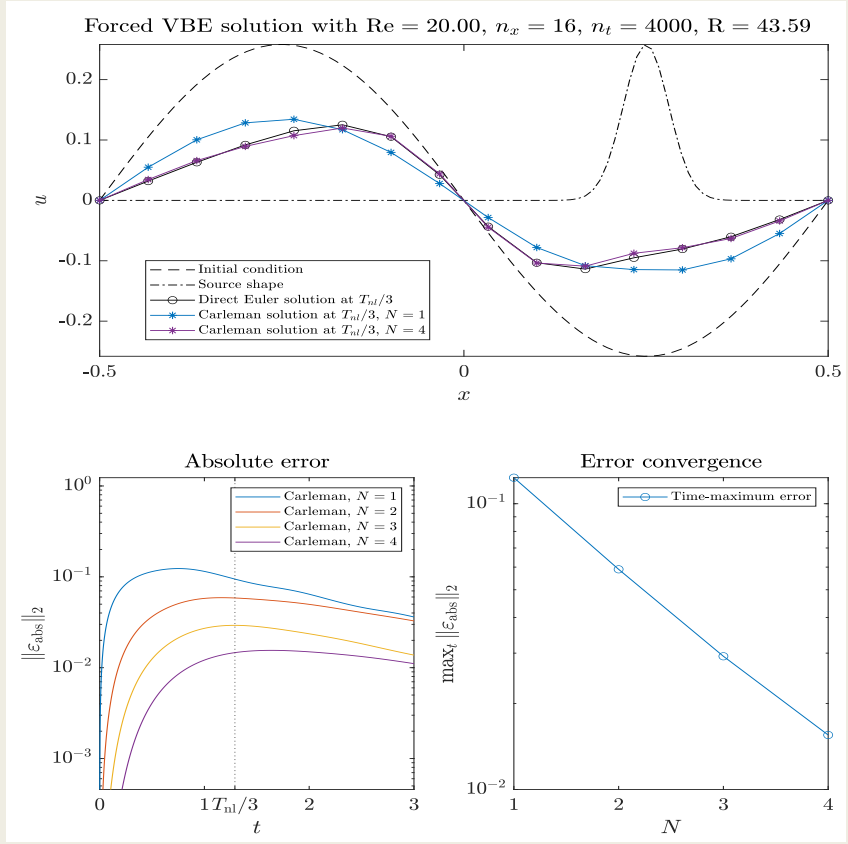


Figure 1: Integration of the forced viscous Burgers equation using Carleman linearization on a classical computer (source code available at https://github.com/hermankolden/CarlemanBurgers). The viscosity is set so that the Reynolds number $\mathrm{Re} = U_0 L_0/\nu = 20$. The parameters $n_x = 16$ and $n_t = 4000$ are the number of spatial and temporal discretization intervals, respectively. The corresponding Carleman convergence parameter is $\mathrm{R} = 43.59$. Top: Initial condition and solution plotted at a third of the nonlinear time $\frac{1}{3}T_{\text{nl}} = \frac{L_0}{3U_0}$. Bottom: l_2 norm of the absolute error between the Carleman solutions at various truncation levels N (left), and the convergence of the corresponding time-maximum error (right).

Lower bounds for higher R

- We also show that *no quantum algorithm* can solve arbitrary nonlinear equations efficiently when $R > \sqrt{2}$.
- To prove this, we show that in the highly nonlinear regime, one can use a hypothetical algorithm to decrease the inner product between two states.
- By making two close by states very orthogonal quickly, one can solve hard problems like Grover search in constant time.

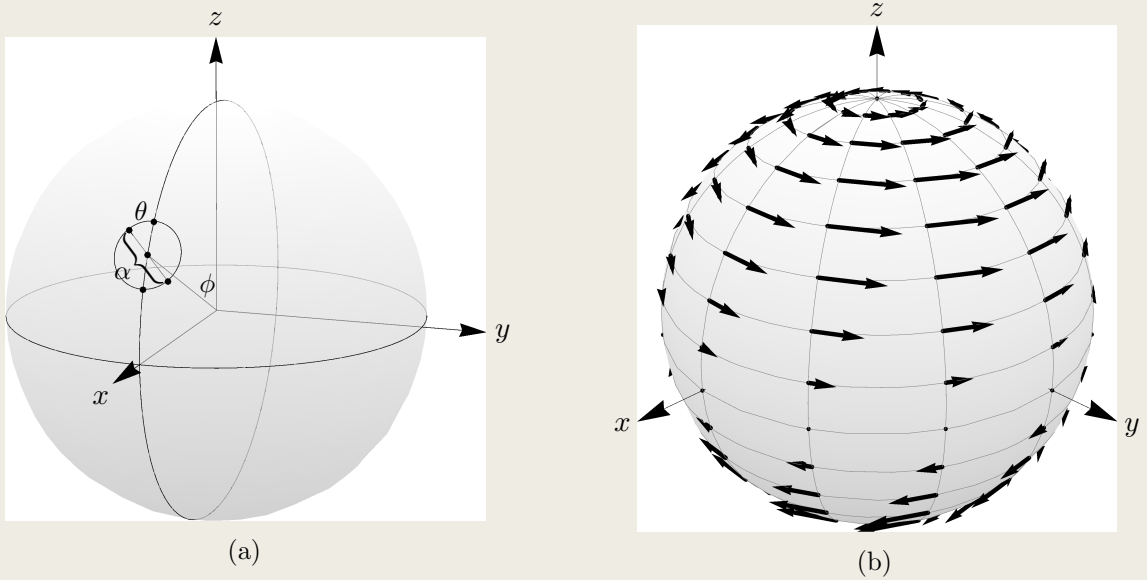


Figure 1: (a) Orientation of states on the Bloch sphere with overlap $\cos\frac{\alpha}{2}$. (b) The flow on the Bloch sphere induced by the Gross-Pitaevskii nonlinearity.

Applications

- Logistic equation

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -r_i u_i + \sum_{j=1}^{n} c_{ij} u_i u_j.$$

- Navier-Stokes

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \beta \mathbf{u} = \nu \nabla^2 \mathbf{u}.$$

- Viscous Burger's equation

$$\partial_t u + u \partial_x u = \frac{1}{\mathrm{Re}} \partial_x^2 u,$$

End