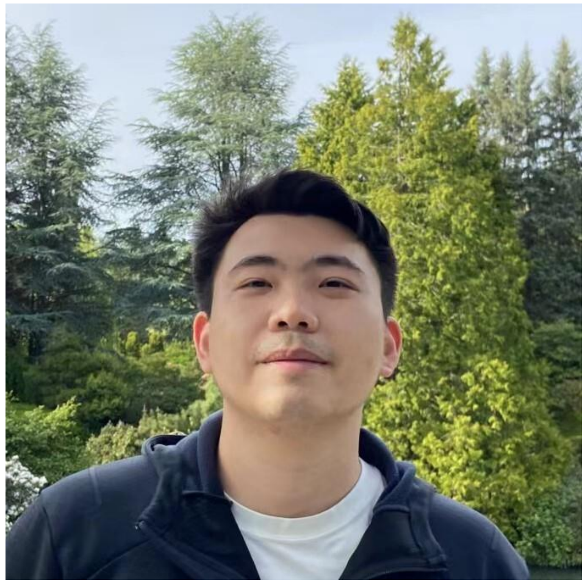
Solving Partial Differential Equations (PDEs) with Quantum Computers

Zhixin (Jack) Song, Spencer H. Bryngelson

Georgia Institute of Technology

09/22 IEEE Quantum Week 2023, Bellevue, WA

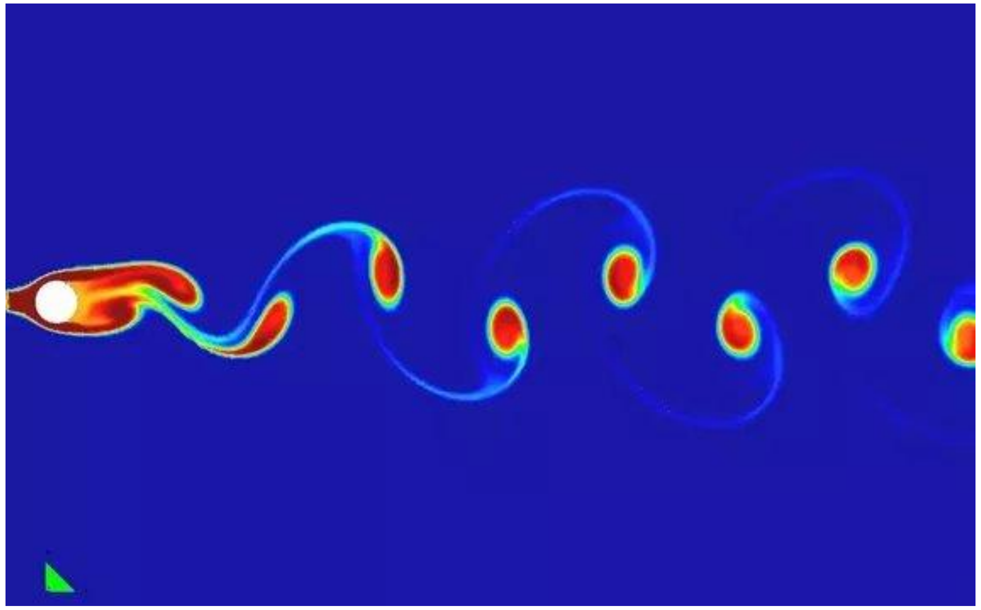
Speakers



Zhixin (Jack) Song
3rd-year Ph.D. Student
zsong300@gatech.edu

Prof. Spencer Bryngelson
PI, shb@gatech.edu

What we do



Computational Fluid Dynamics (CFD)

+

Quantum Computing (QC)

https://comp-physics.group/

Timeline

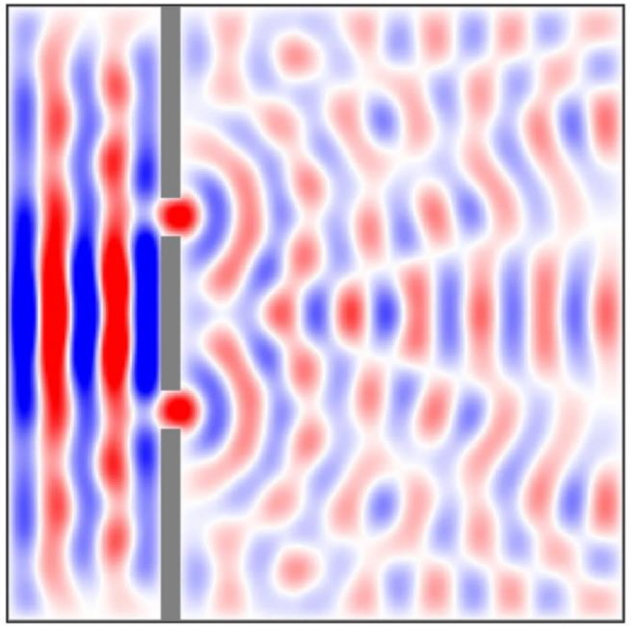
- Session 1 (10-11:30 am)
  - Introduction (20 min)
  - Classical Solution (10 min)
  - Quantum Solution (30 min)
  - Notebook on Hamiltonian Simulation (30 min)
- Session 2 (1-2:30 pm)
  - Near-term QC (30 min)
  - Notebook on VQA (40 min)
  - Open discussion (20 min)

Timeline

- Session 1 (10-11:30 am)
    - Introduction (20 min)
    - Classical Solution (10 min)
    - Quantum Solution (30 min)
    - Notebook on Hamiltonian Simulation (30 min)
- Session 2 (1-2:30 pm)
    - Near-term QC (30 min)
    - Notebook on VQA (40 min)
    - Open discussion (20 min)

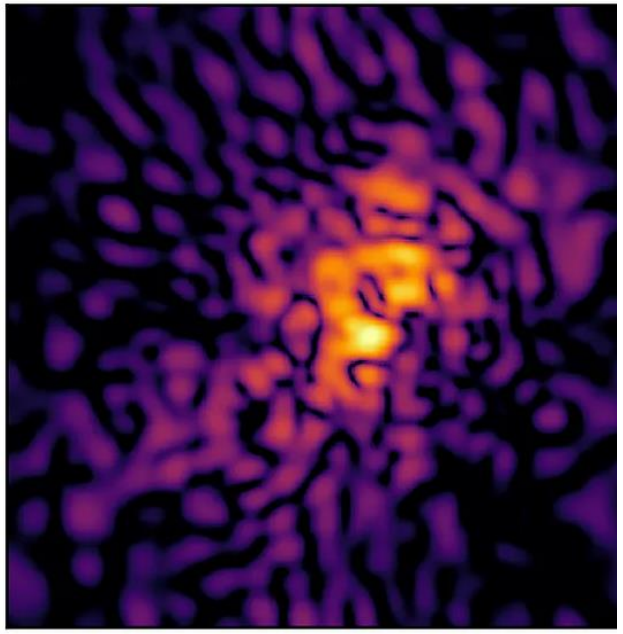
What is a PDE?

An equation to solve with partial derivatives.



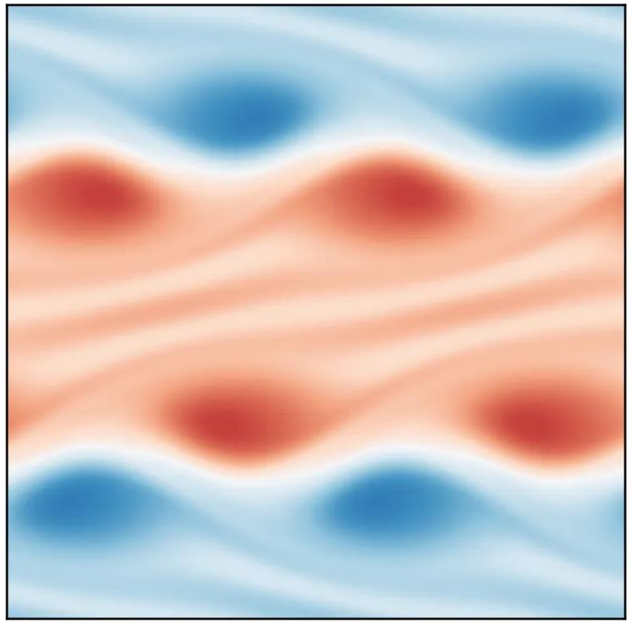
Wave equation

$$\frac{\partial^2 U}{\partial t^2} = c^2 \nabla^2 U$$



Schrödinger equation

$$i\hbar \frac{\partial}{\partial t}\psi = -\frac{\hbar^2}{2m}\nabla^2 \psi + mV\psi$$



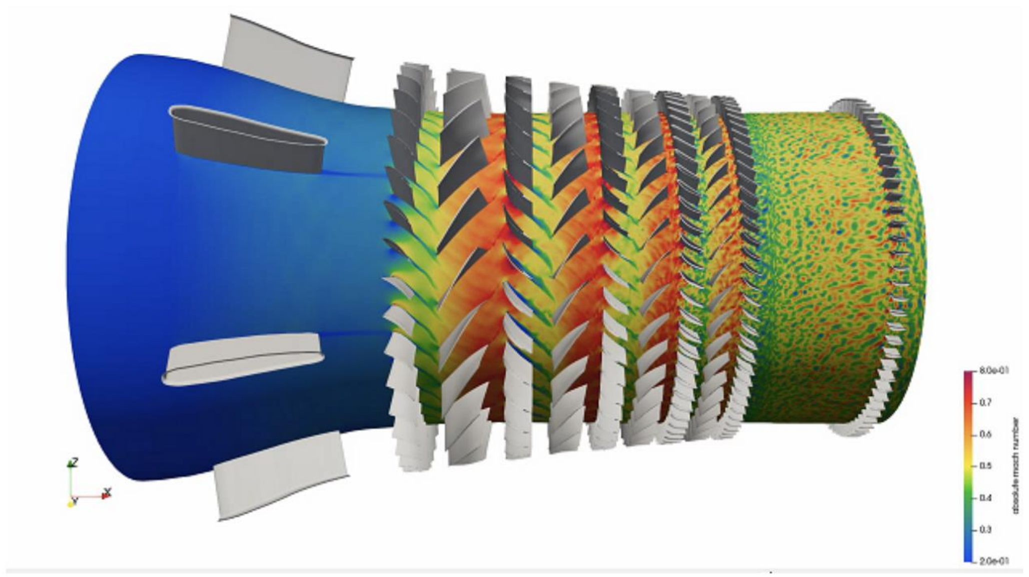
Navier-Stokes equations

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\,\mathbf{v} = \nu \nabla^2 \mathbf{v} - \nabla P$$

Source: https://philip-mocz.medium.com/

Why quantum computing?

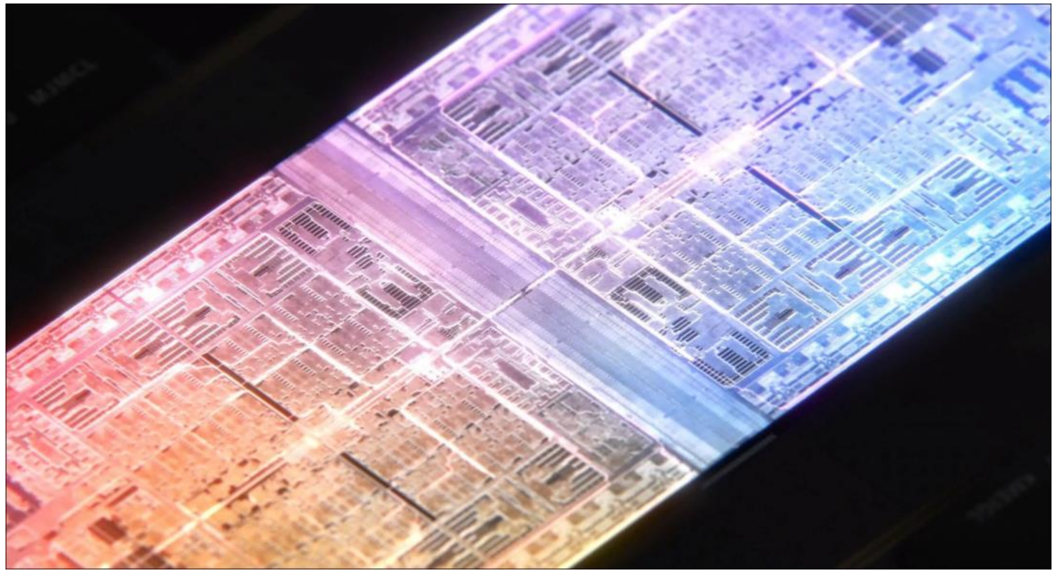
Solving such problems in large scale is expensive! Each high resolution CFD simulation could cost millions of dollars and produce ~tons of CO_2.



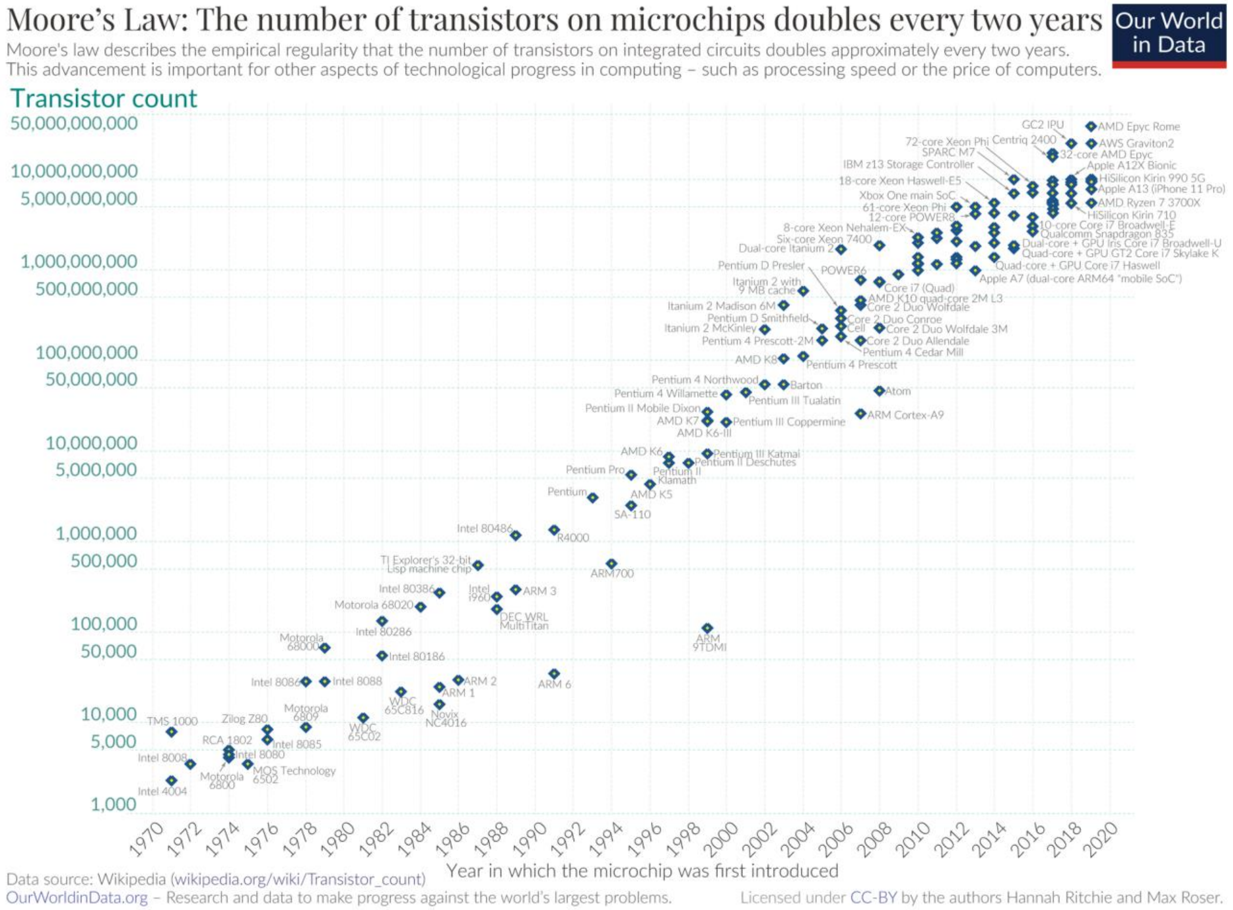
Example CFD simulation using 5.3 billion nodes (~32 qubits)

Gerstenberger, A. (2021). 3d cfd in-situ co-processing for turbomachinery design. In *ISAV'21: In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization*.

Why quantum computing?



Apple's 5nm process M1 Ultra chip contains 114 billion transistors. As the size of transistors gradually shrinks to a few atoms, it will no longer be a reliable bit due to quantum tunneling.



Moore's Law: The number of transistors on microchips doubles every two years

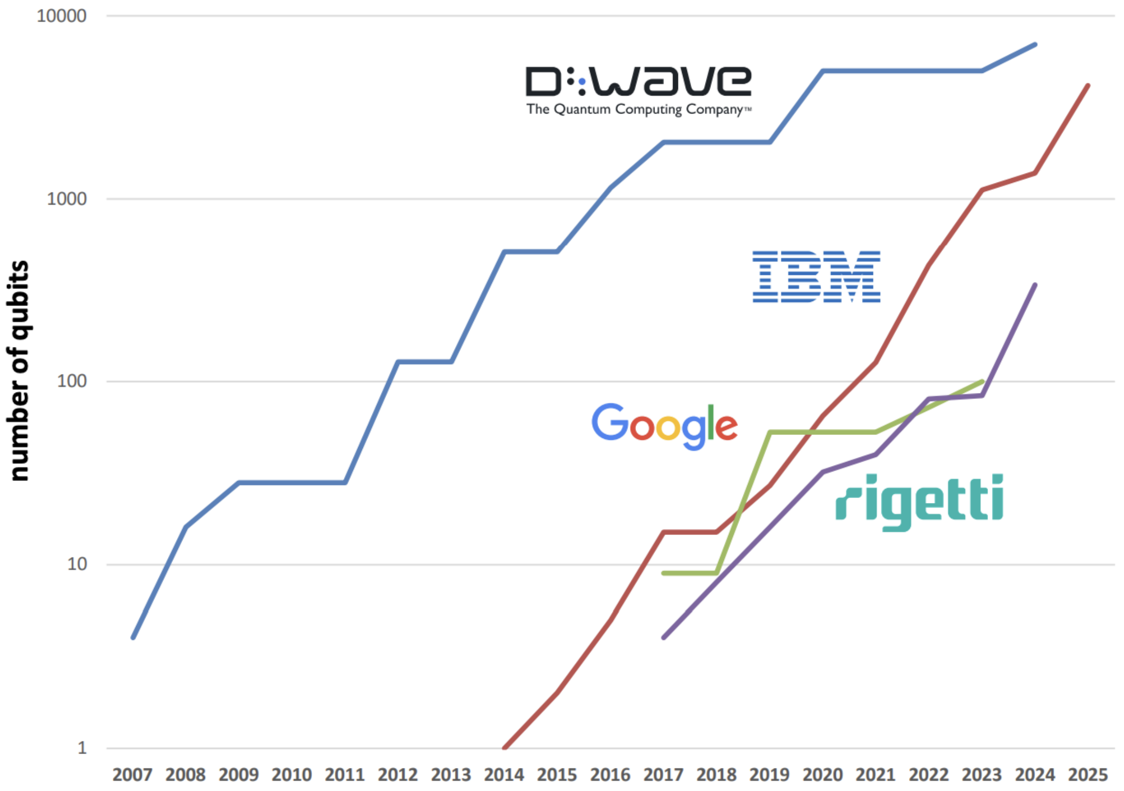
Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

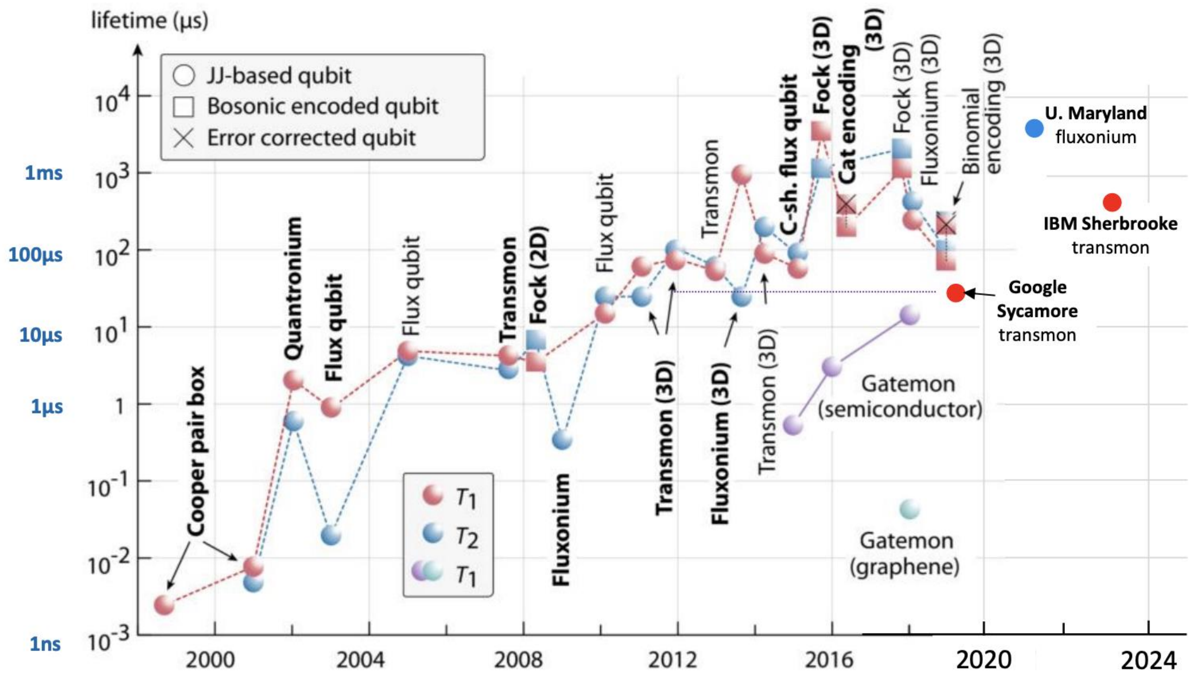
Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)
OurWorldinData.org – Research and data to make progress against the world's largest problems.
Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

https://www.techspot.com/news/93755-exploring-apple-m1-ultrafusion-chip-interconnect.html

Why quantum computing?

- Computational space doubles every time you add a qubit

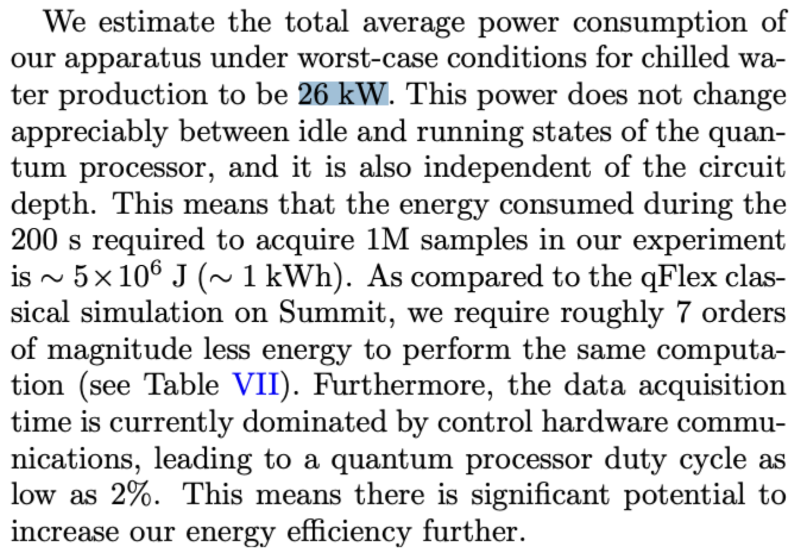




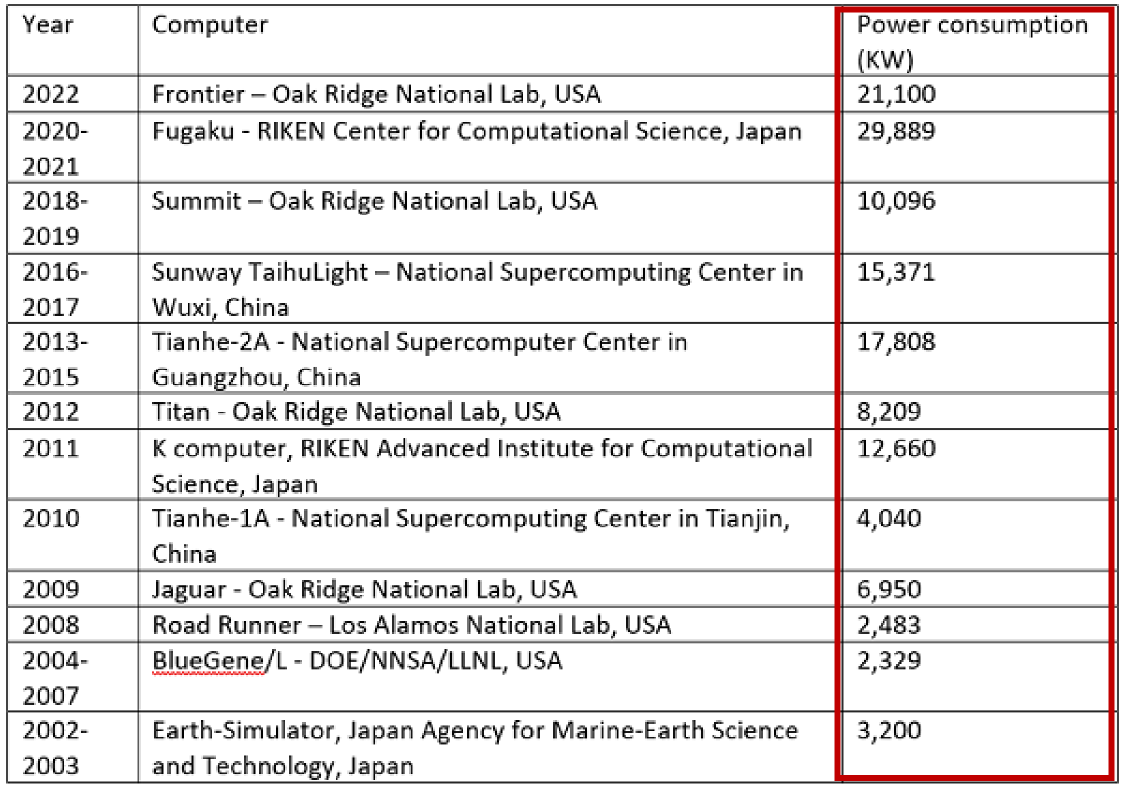
Ezratty, O. (2023). Is there a Moore's law for quantum computing?. *arXiv preprint arXiv:2303.15547*.

Why quantum computing?

- No heat created (in principle) according to Landauer's principle
- Energy consumption is several orders of magnitude lower



We estimate the total average power consumption of our apparatus under worst-case conditions for chilled water production to be 26 kW. This power does not change appreciably between idle and running states of the quantum processor, and it is also independent of the circuit depth. This means that the energy consumed during the 200 s required to acquire 1M samples in our experiment is $\sim 5 \times 10^6$ J (~ 1 kWh). As compared to the qFlex classical simulation on Summit, we require roughly 7 orders of magnitude less energy to perform the same computation (see Table VII). Furthermore, the data acquisition time is currently dominated by control hardware communications, leading to a quantum processor duty cycle as low as 2%. This means there is significant potential to increase our energy efficiency further.



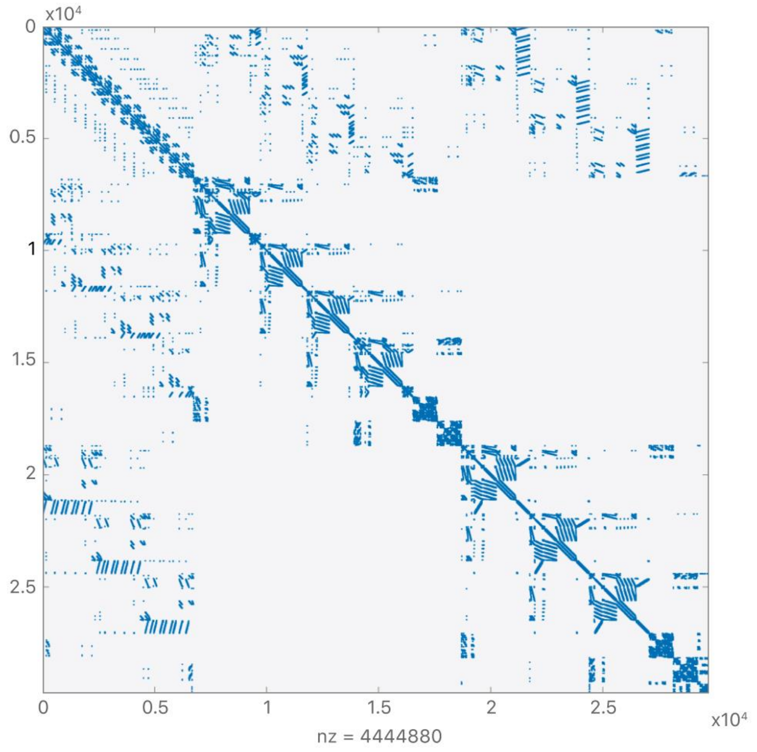
Year	Computer	Power consumption (KW)
2022	Frontier – Oak Ridge National Lab, USA	21,100
2020-2021	Fugaku - RIKEN Center for Computational Science, Japan	29,889
2018-2019	Summit – Oak Ridge National Lab, USA	10,096
2016-2017	Sunway TaihuLight – National Supercomputing Center in Wuxi, China	15,371
2013-2015	Tianhe-2A - National Supercomputer Center in Guangzhou, China	17,808
2012	Titan - Oak Ridge National Lab, USA	8,209
2011	K computer, RIKEN Advanced Institute for Computational Science, Japan	12,660
2010	Tianhe-1A - National Supercomputing Center in Tianjin, China	4,040
2009	Jaguar - Oak Ridge National Lab, USA	6,950
2008	Road Runner – Los Alamos National Lab, USA	2,483
2004-2007	BlueGene/L - DOE/NNSA/LLNL, USA	2,329
2002-2003	Earth-Simulator, Japan Agency for Marine-Earth Science and Technology, Japan	3,200

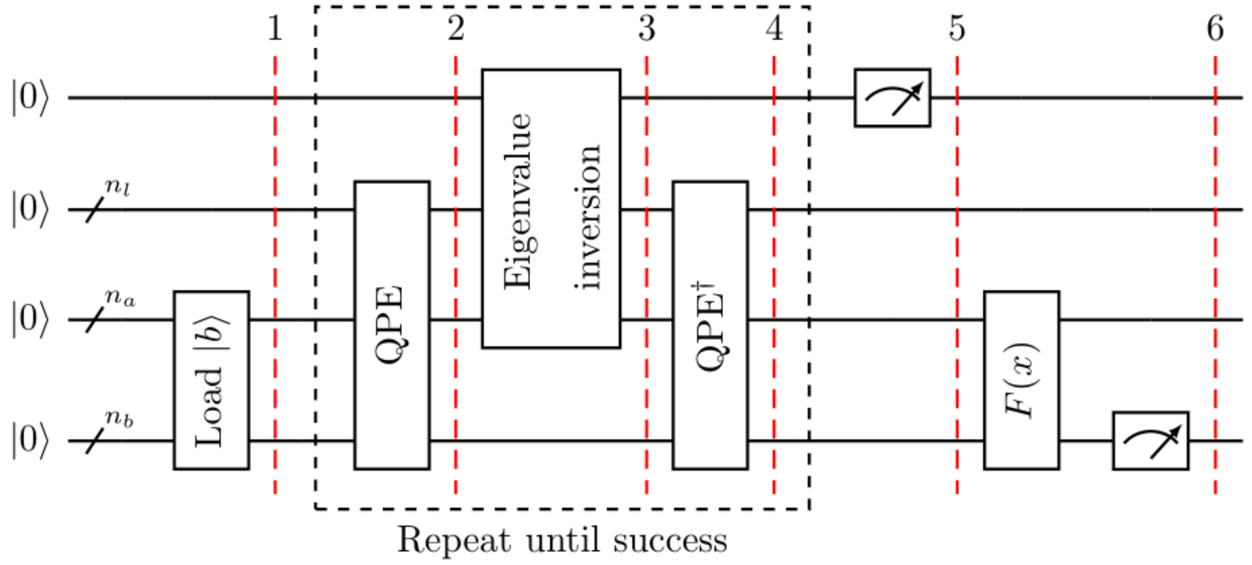
https://insidehpc.com/2023/06/the-energy-advantage-of-quantum-computers/

Liu, J., Jiang, H., & Shen, Z. J. M. (2023). Potential Energy Advantage of Quantum Economy. *arXiv preprint arXiv:2308.08025*.

Why quantum computing?

- The core of fast CFD simulation is a large sparse linear solver
- Efficient quantum algorithms exist for solving linear systems



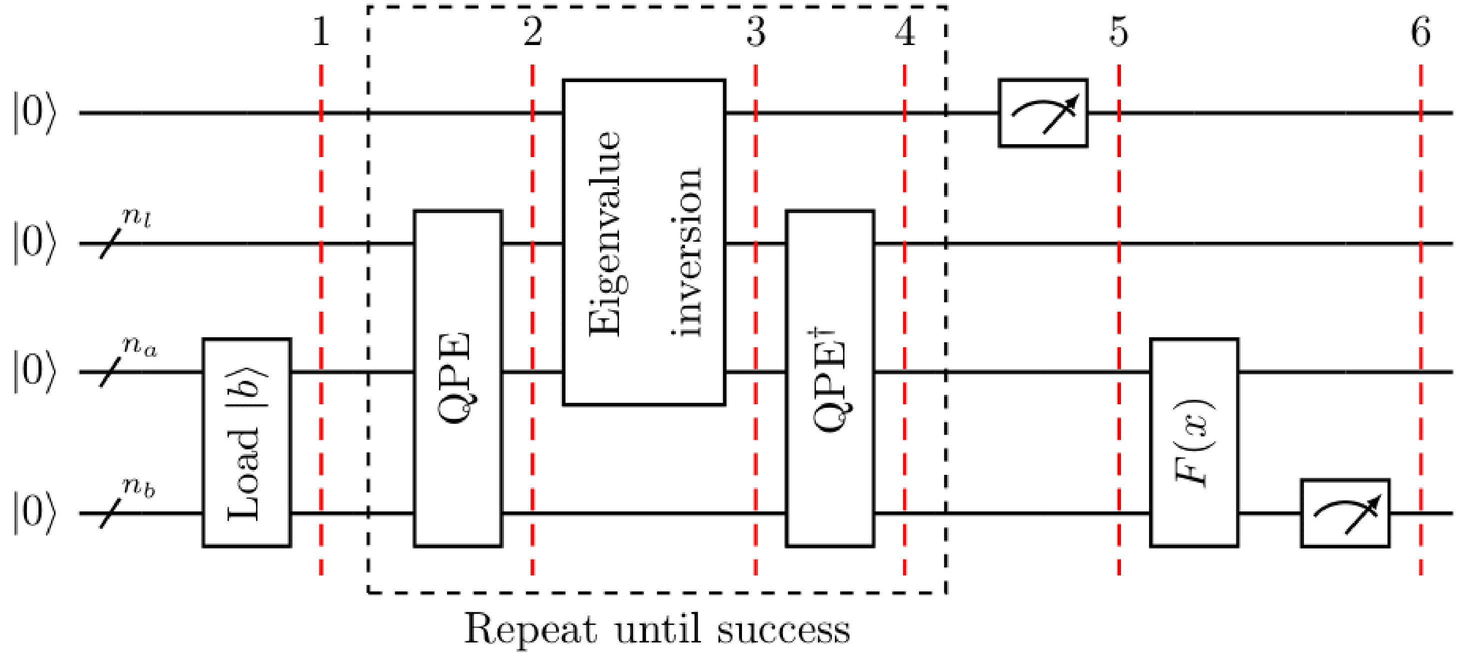


https://developer.apple.com/documentation/accelerate/sparse_solvers/

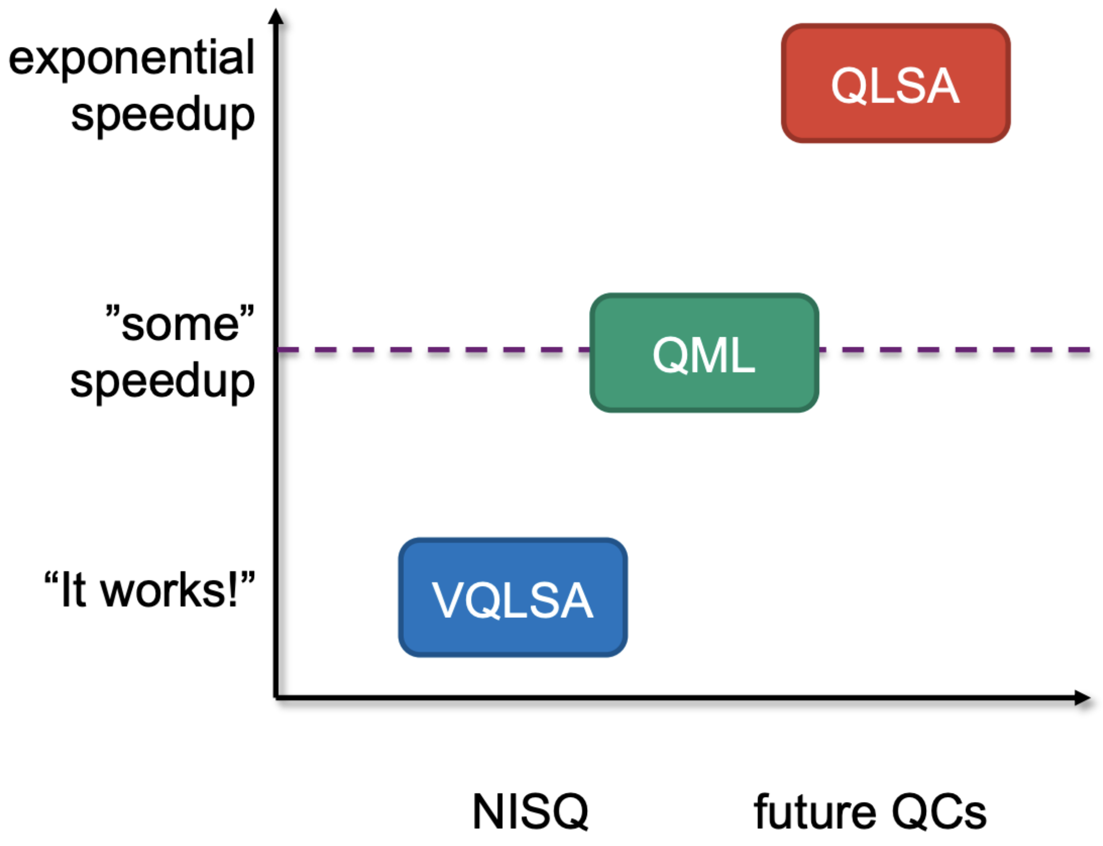
Harrow, A. W., Hassidim, A., & Lloyd, S. (2009). Quantum algorithm for linear systems of equations. *Physical review letters*, *103*(15), 150502.

Is that enough?

- HHL (09) claims exponential speed-up $O(s^2\kappa^2 \log(N)/\epsilon)$
- Textbook Gaussian elimination requires $O(N^3)$. Is this the best a classical solver can do?
- What's the cost of SPAM? State tomography will ruin all the speed up



What should we aim for?



At least a ***quadratic speedup*** is needed considering the clock speed difference.

Classical: GHz
Quantum: kHz-MHz

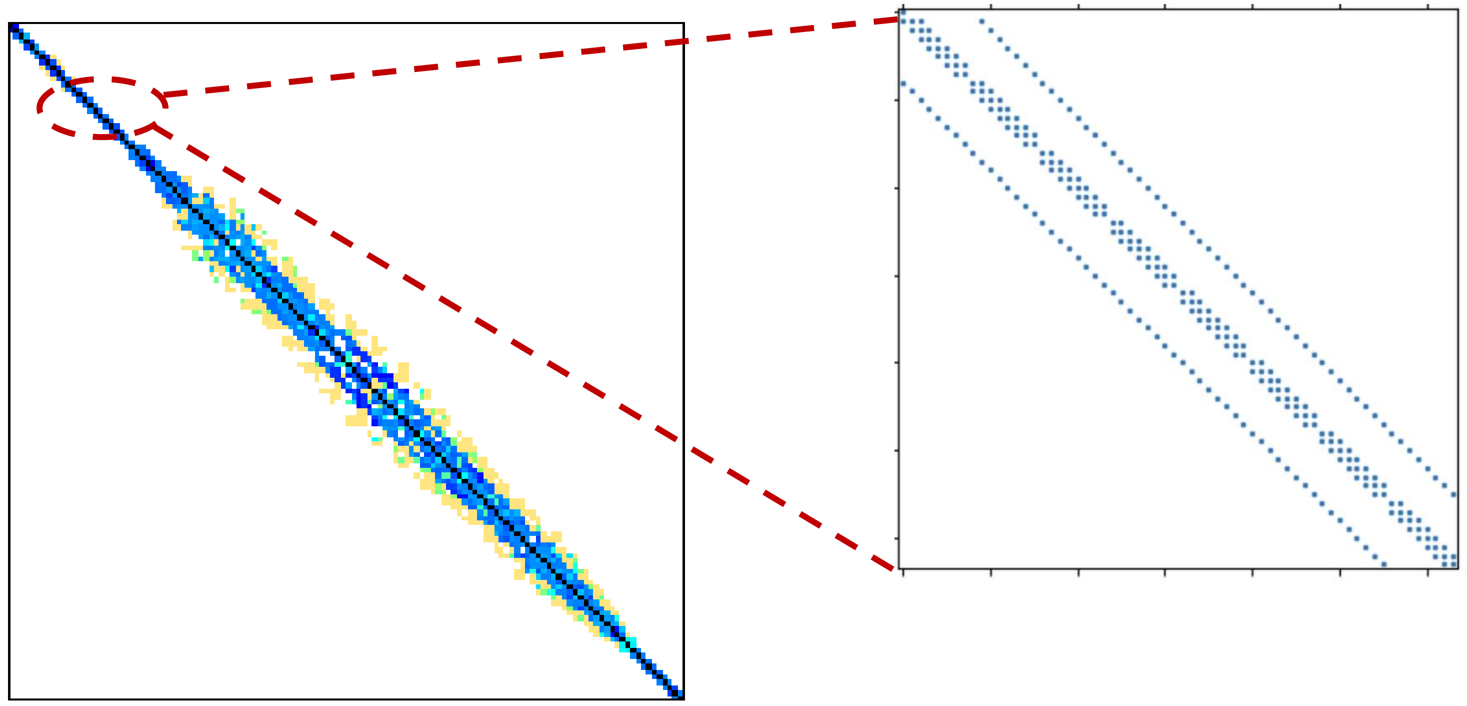
Source: https://tinyurl.com/qc-clockspeed

Timeline

- Session 1 (10-11:30 am)
    - Introduction (20 min)
    - Classical Solution (10 min)
    - Quantum Solution (30 min)
    - Notebook on Hamiltonian Simulation (30 min)
- Session 2 (1-2:30 pm)
    - Near-term QC (30 min)
    - Notebook on VQA (40 min)
    - Open discussion (20 min)

Sparse linear system

- Most matrices arising from real applications are sparse



Example sparse A matrix in CFD simulations

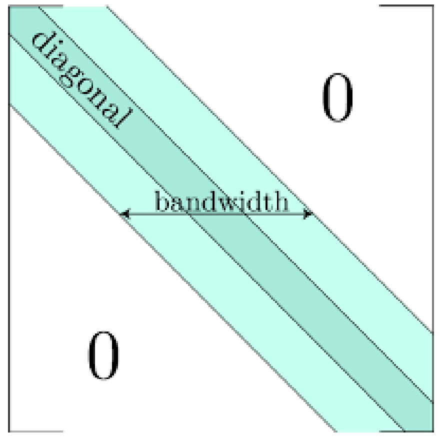
- Sparse matrix collection https://sparse.tamu.edu

Jargons

- **Sparsity:** number of elements are non-zero
- **Condition number:** measures how much the output value of the function can change for a small change in the input argument

$$\kappa(A) = \|A\| \cdot \left\|A^{-1}\right\|$$

- **Bandwidth:** the maximum number of diagonals on either side of the main diagonal that contain non-zero elements



Zoo

- Direct methods
  - LU decomposition (Gaussian Elimination) $O(N^3)$
- Iterative methods
  - Jacobi/Gauss-Seidel $O(N^2)$
  - Conjugate Gradient (CG) $O(N\sqrt{\kappa})$
  - Multigrid $O(N)$

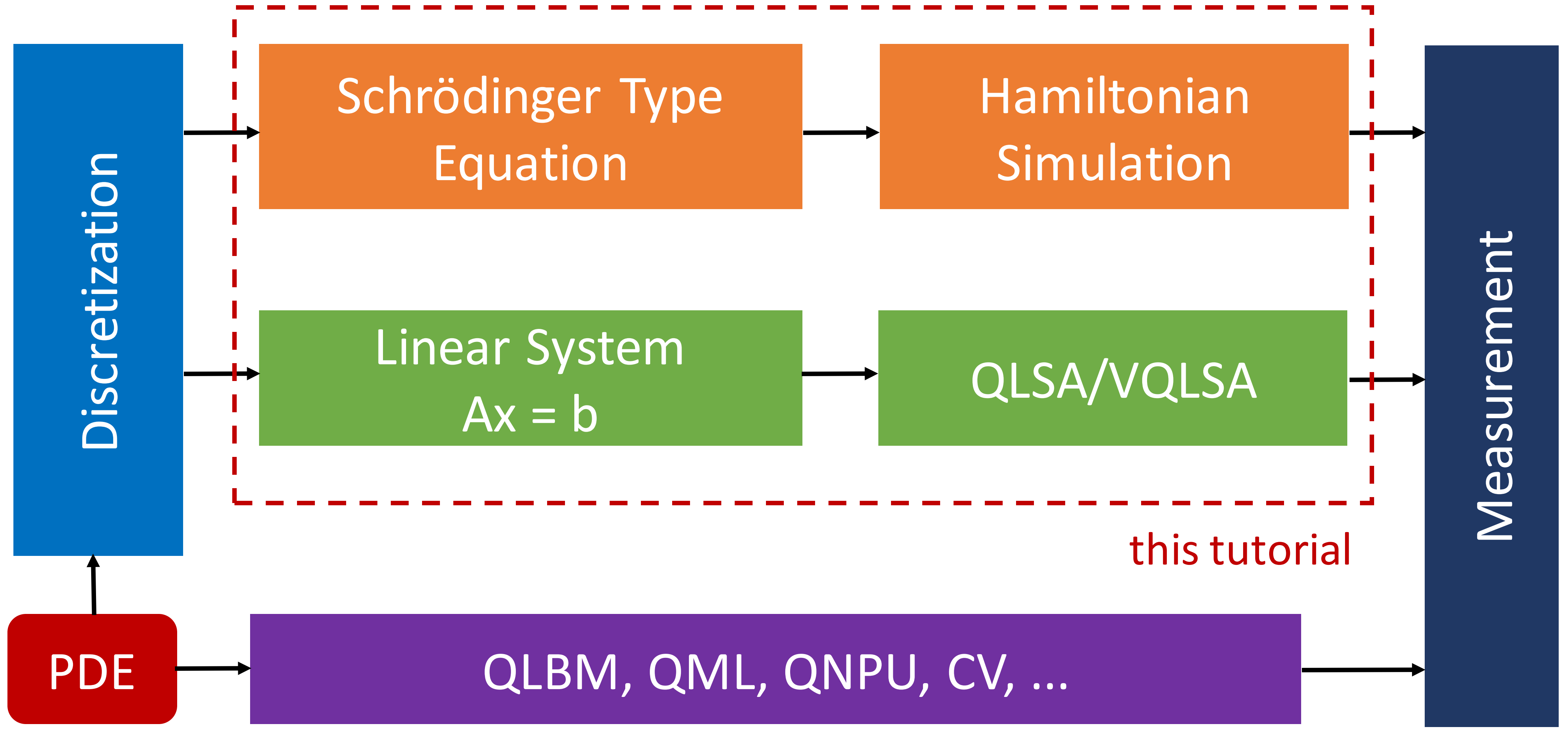
SciPy PETSc TRILINOS

https://petsc.org/release/overview/linear_solve_table/

Timeline

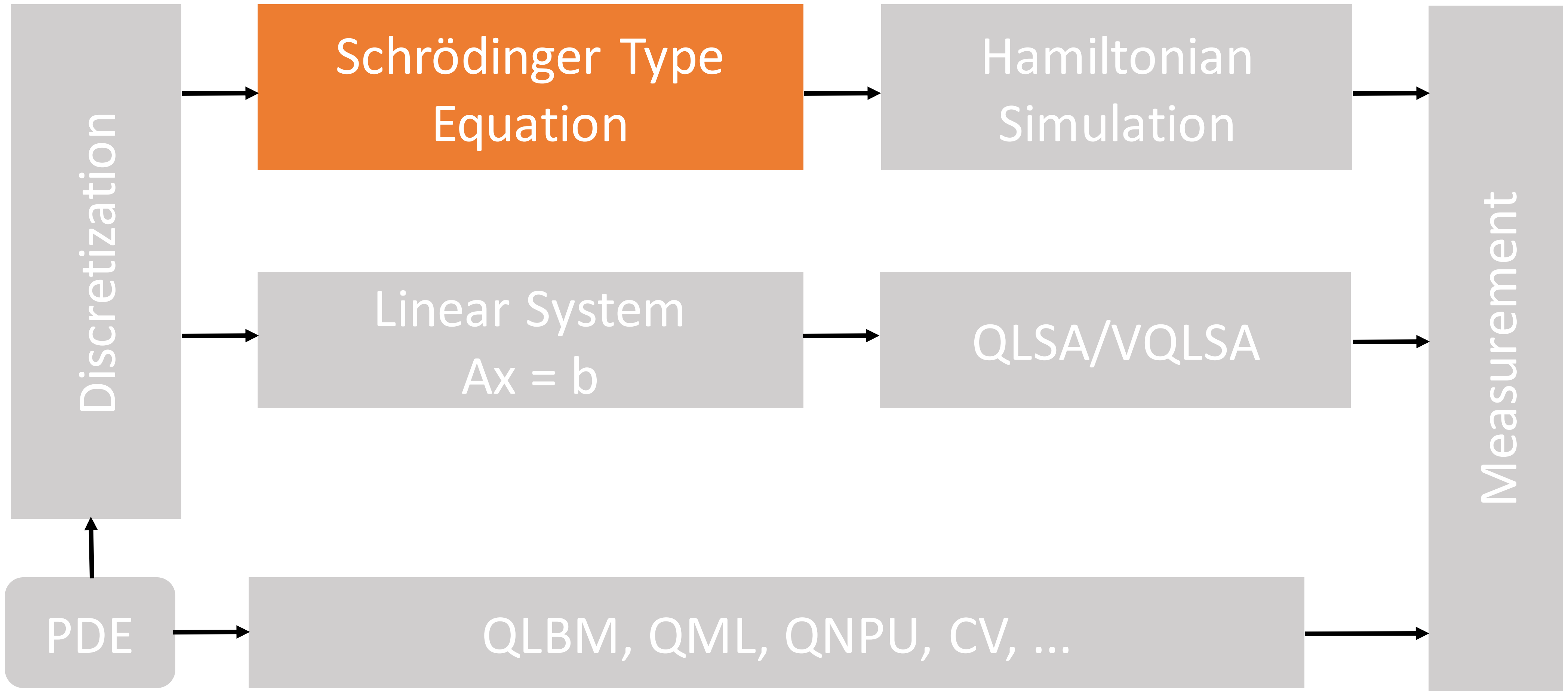
- Session 1 (10-11:30 am)
  - Introduction (20 min)
  - Classical Solution (10 min)
  - Quantum Solution (30 min)
  - Notebook on Hamiltonian Simulation (30 min)
- Session 2 (1-2:30 pm)
  - Near-term QC (30 min)
  - Notebook on VQA (40 min)
  - Open discussion (20 min)

Quantum Algorithms for solving PDEs

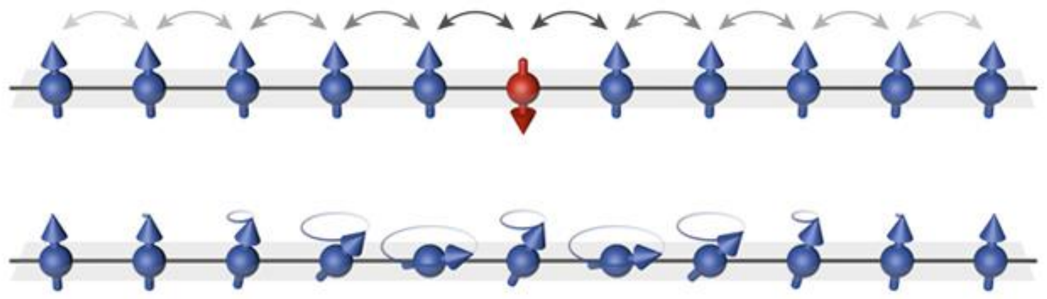


Pesah, A. (2020). Quantum algorithms for solving partial differential equations. *University College London*.

Quantum Algorithms for solving PDEs



Schrödinger Equation



- Goal is to solve $$\frac{d\psi}{dt} = iH\psi$$

- Can be achieved by efficiently implementing unitary operator

$$|\psi\rangle = e^{-iHt}\,|\psi_0\rangle$$

- The most natural type of PDE to solve on a quantum computer
- Some PDEs can be converted into this scheme

Wave Equation

- Goal: $\frac{\partial^2 f}{\partial t^2} = -\Delta f$
- Observe derivative of the Schrödinger's equation $\frac{d^2\psi}{dt^2} = -H^2\psi$

Laplacian

- Hermitian matrix

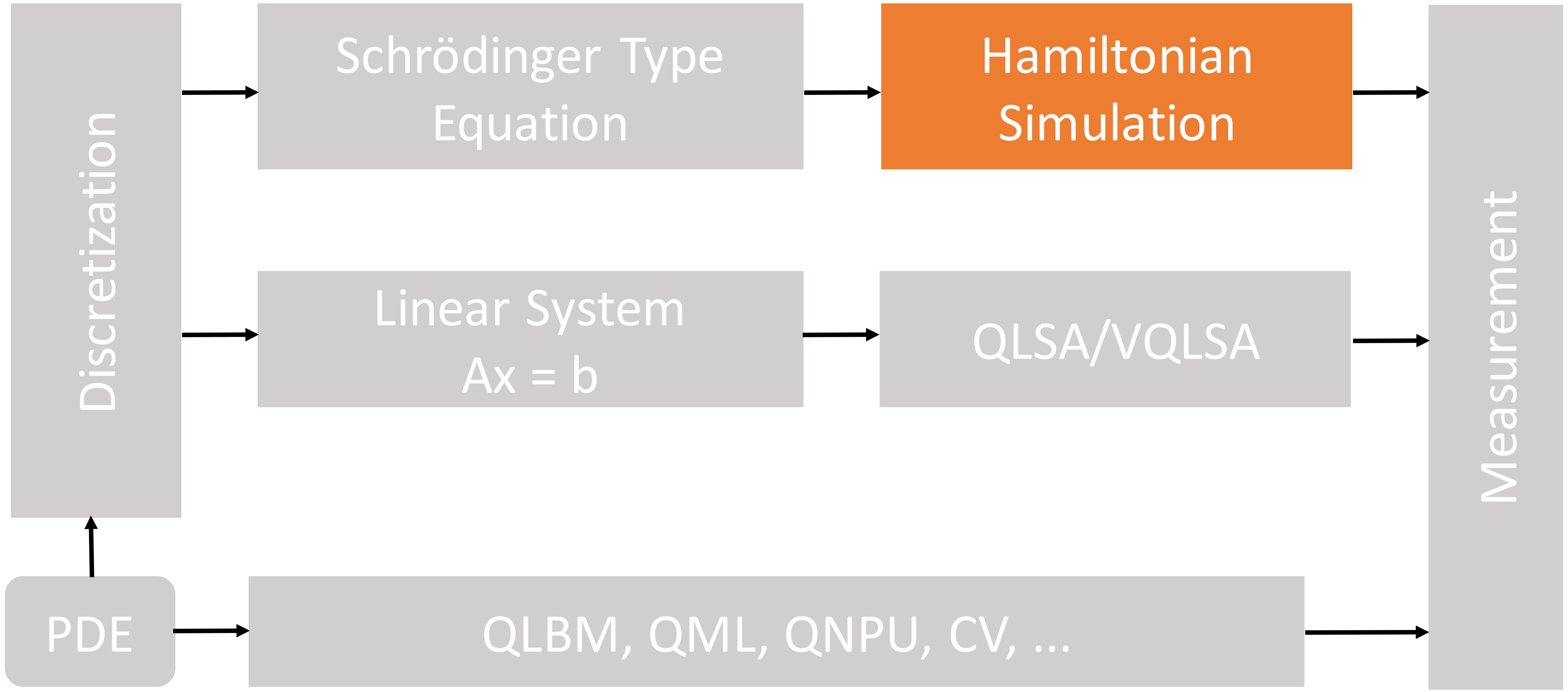
$$H = \begin{pmatrix} 0 & B \\ B^\dagger & 0 \end{pmatrix} \Rightarrow H^2 = \begin{pmatrix} \boxed{BB^\dagger} & 0 \\ 0 & B^\dagger B \end{pmatrix}$$

- Can be achieved by efficiently implement the unitary operator

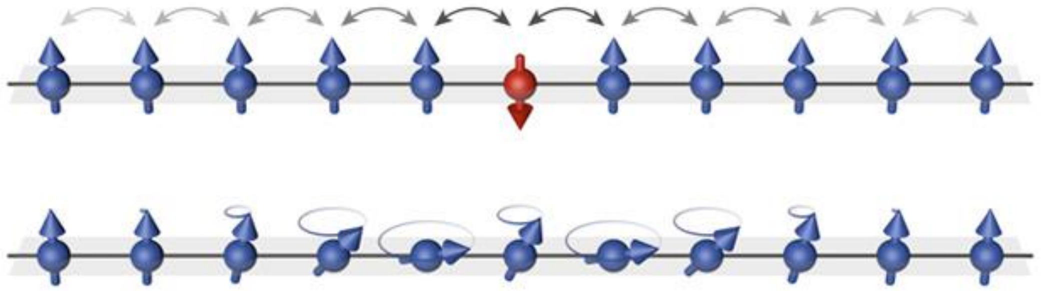
$$|\psi\rangle = e^{-iHt}\,|\psi_0\rangle$$

- Not all PDEs can be written as a Schrödinger's equation!!

Quantum Algorithms for solving PDEs



Hamiltonian Simulation



Given: Hamiltonian

Goal: find algorithm to approximate $\left\|U - e^{iHt}\right\| \leq \epsilon$

- Decompose into local Hamiltonians

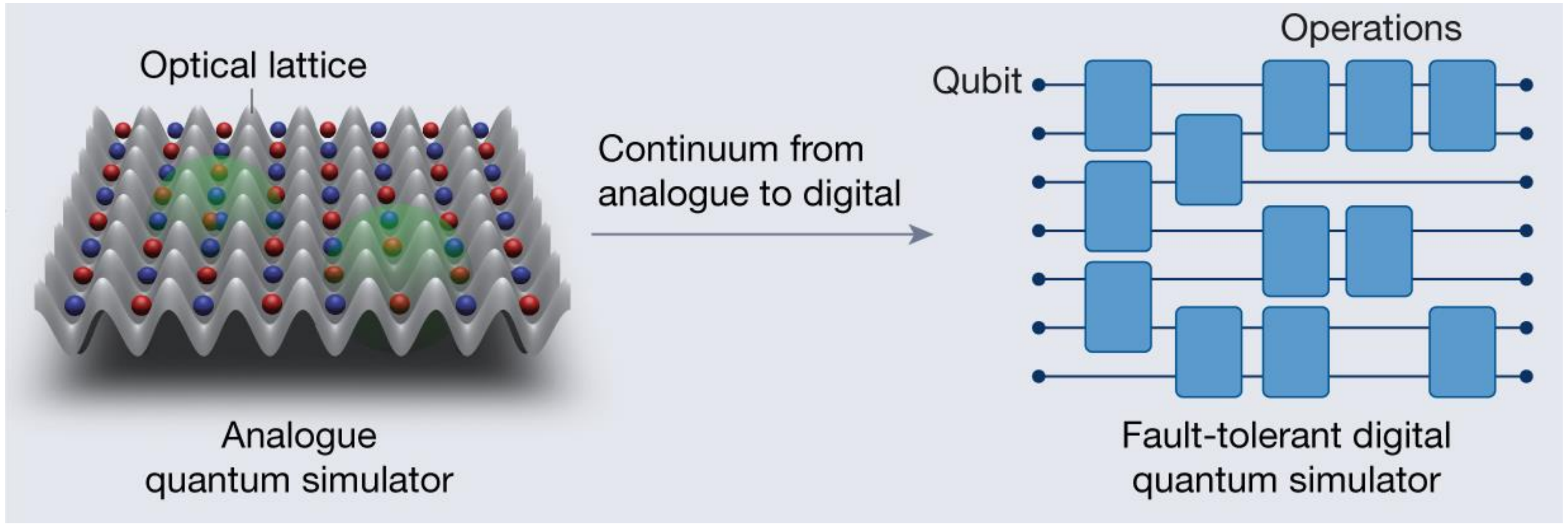
$$H = \sum_k H_k$$

- Suzuki-Trotter decomposition

$$e^{-iHt} = \left(e^{-iH\tau}\right)^r = \left(\prod_{k=1}^{L} e^{-iH_k\tau} + O\left(\tau^2\right)\right)^r = \left(\prod_{k=1}^{L} e^{-iH_k\tau}\right)^r + O\left(\frac{t^2}{r}\right)$$

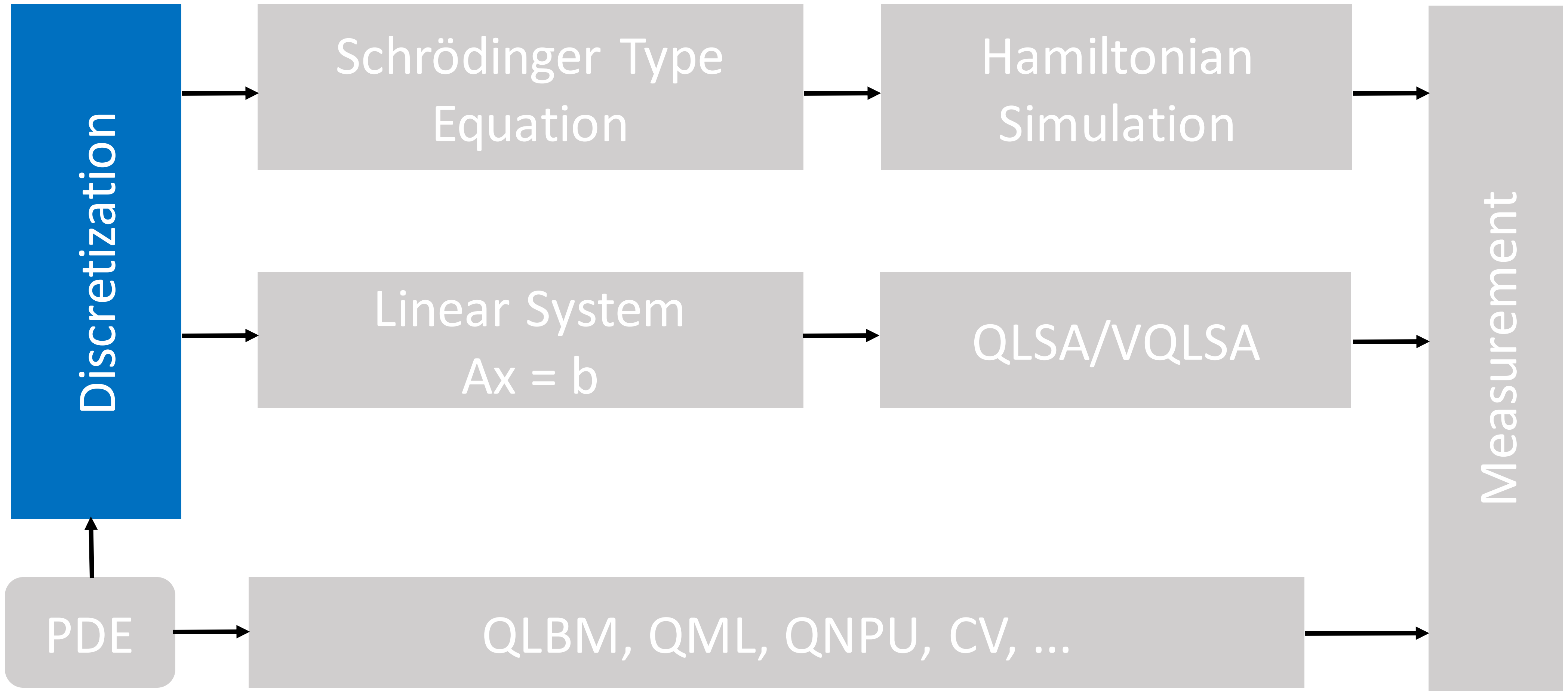
Analog V.S. Digital

- The target Hamiltonian is close to the system Hamiltonian itself
- So far, most meaningful results are from analog simulators
- Digital simulation (gate based) is difficult due to hardware limitations


Optical lattice
Analogue quantum simulator
Continuum from analogue to digital
Operations
Qubit
Fault-tolerant digital quantum simulator

Daley, A. J., Bloch, I., Kokail, C., Flannigan, S., Pearson, N., Troyer, M., & Zoller, P. (2022). Practical quantum advantage in quantum simulation. *Nature*, *607*(7920), 667-676.

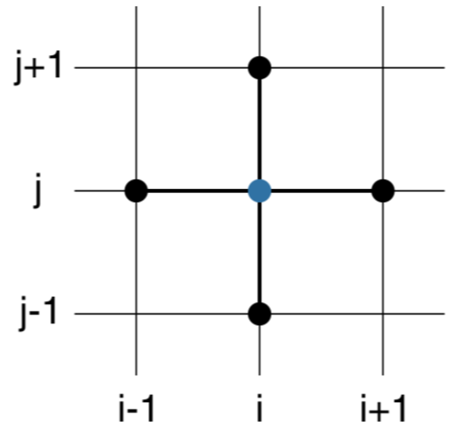
Quantum Algorithms for solving PDEs



Example

- Discretize the 1D spatial domain into grid of n points and approximate derivative via central difference (CDS)

$$\frac{\partial^2 \phi(x)}{\partial x^2} \approx \frac{\phi_{i-1} - 2\phi_i + \phi_{i+1}}{h^2}$$



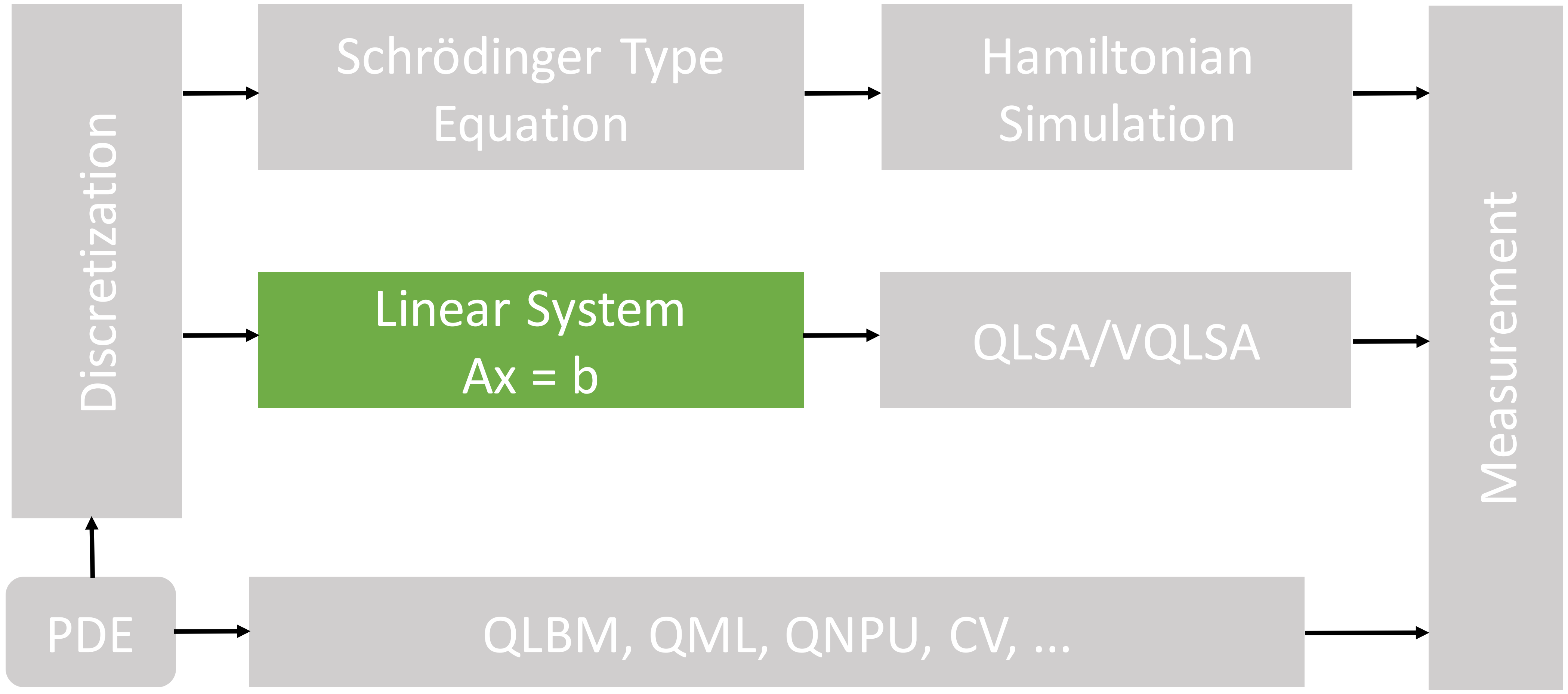
- 2D Laplacian

$$\nabla^2 \phi_{i,j} = \frac{1}{h^2} \left(\phi_{i-1,j} + \phi_{i+1,j} + \phi_{i,j-1} + \phi_{i,j+1} - 4\phi_{i,j}\right)$$

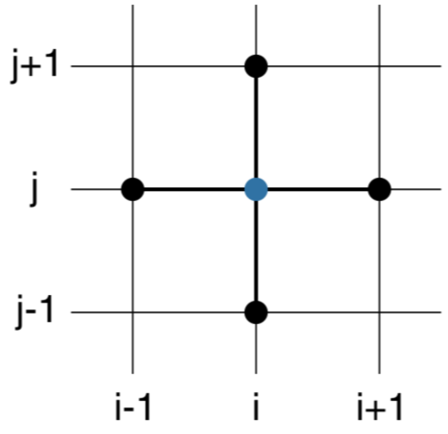
- For many equations, achieving a discretization error less than ϵ requires

$$N = \mathcal{O}\left(\text{poly}\left(\frac{1}{\epsilon^d}\right)\right)$$

Quantum Algorithms for solving PDEs



Linear system Ax=b



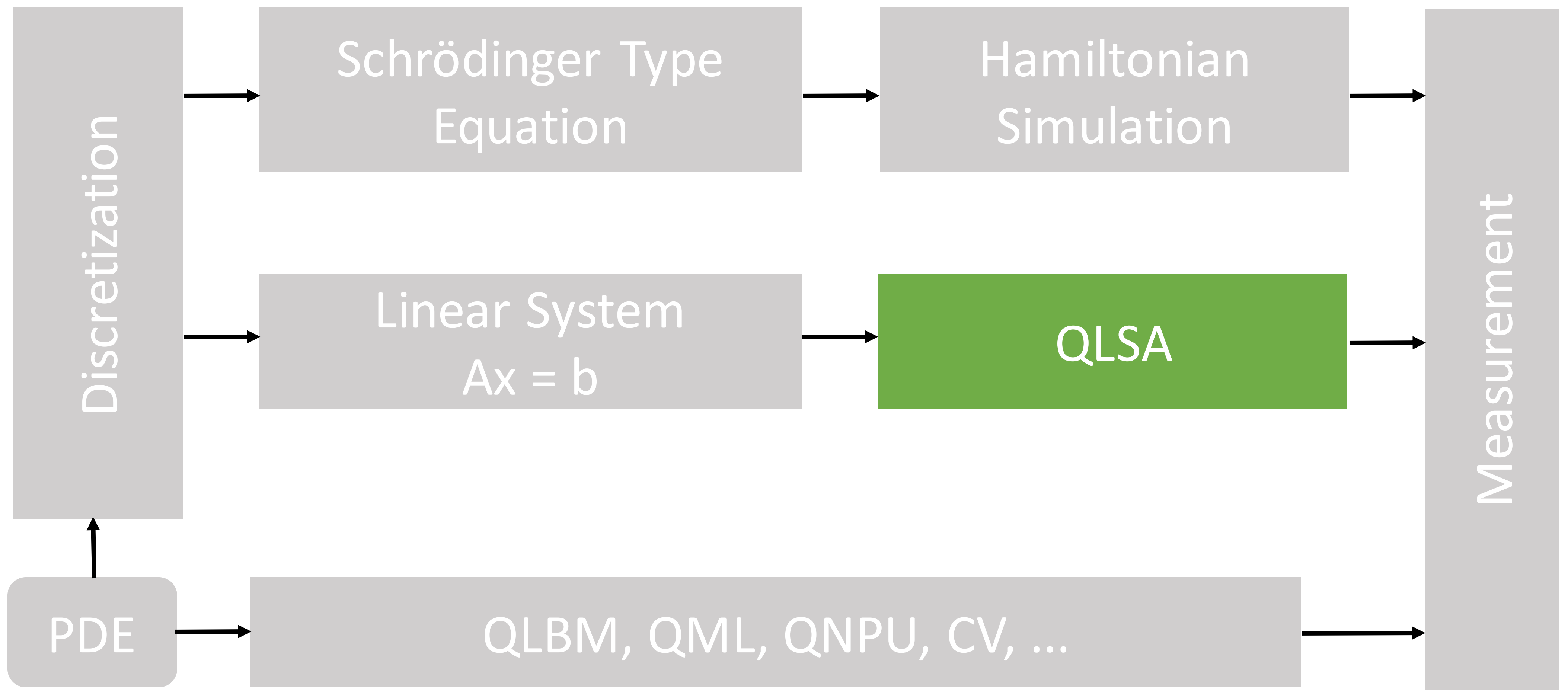
- For example, let's consider a 1D Poisson equation

$$\Delta\phi(x) = f(x), \quad \textbf{with} \quad \phi(a) = \phi_a, \phi(b) = \phi_b.$$

- Apply the central difference scheme (CDS), we get

$$A = \begin{bmatrix} -2 & 1 & 0 & \cdots & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -2 \end{bmatrix}, b = \begin{bmatrix} h^2 f_1 - \phi_a \\ h^2 f_2 \\ \vdots \\ h^2 f_{n-1} \\ h^2 f_n - \phi_b \end{bmatrix}.$$

Quantum Algorithms for solving PDEs

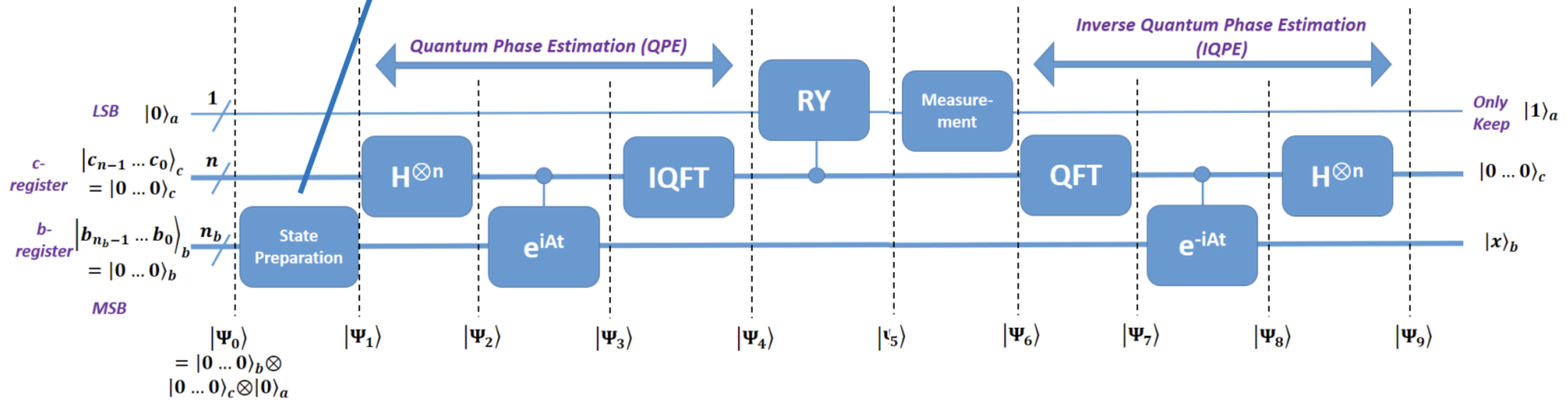


HHL Algorithm

$O(s^2\kappa^2 \log(N)/\epsilon)$

- **Goal:** prepare the quantum state $|x\rangle$ that solves $Ax = b$

$$A = \sum_{j=0}^{2^n-1} \lambda_j |u_j\rangle\langle u_j|, \qquad \boxed{b = \sum_{j=0}^{2^n-1} b_j |u_j\rangle,} \qquad |x\rangle = A^{-1}|b\rangle = \sum_{j=0}^{2^n-1} \lambda_j^{-1} b_j |u_j\rangle$$



https://homepage.tudelft.nl/y95n9/files/presentations/qc4pde2022.pdf
Morrell Jr, H. J., Zaman, A., & Wong, H. Y. (2021). Step-by-step hhl algorithm walkthrough to enhance the understanding of critical quantum computing concepts. *arXiv preprint arXiv:2108.09004*.

State preparation

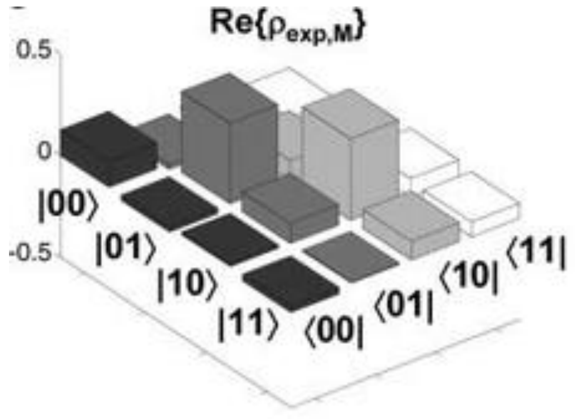
- General states cannot be prepared efficiently, not even approximated!
- Requires $O(N)$ gates using uniformly controlled rotations
- Certain states of the form $|\psi\rangle = \sum_i \sqrt{p_i}|i\rangle$ can be prepared efficiently, e.g., using quantum GANs with $\mathcal{O}(\mathrm{polylog}(N))$ gates

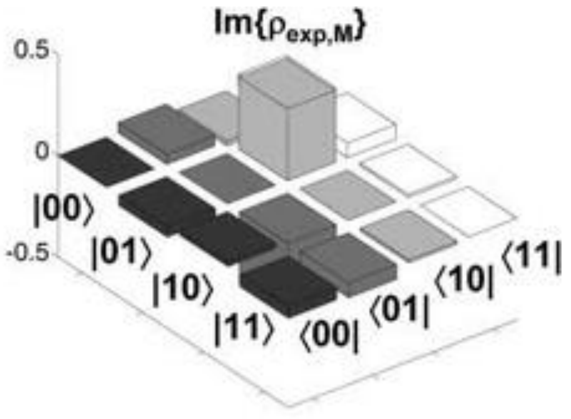
$$\frac{\partial^2 \log(p(x))}{\partial x^2} < 0$$ such as normal distribution

- Possible to reduce time complexity by adding ancillary qubits

Measurement

- Access to full quantum solution encoded on amplitudes require Quantum State Tomography (QST)
- This requires $\mathcal{O}(4^n)$ of state sampling + MLE post-processing
- Cost few hours to reconstruct a ~20 qubit density matrix and this kill the exponential speed-up!





Measurement

- The global solution cannot be easily extracted in general
- Many functions of a state that can be extracted efficiently
  - Inner product estimation (SWAP test)
  - Small set of amplitudes (amplitude estimation)
- Measure any functional requires $\mathcal{O}(\mathrm{ploy}(1/\epsilon))$ repetitions
- This cost is unavoidable due to the nature of quantum computing!

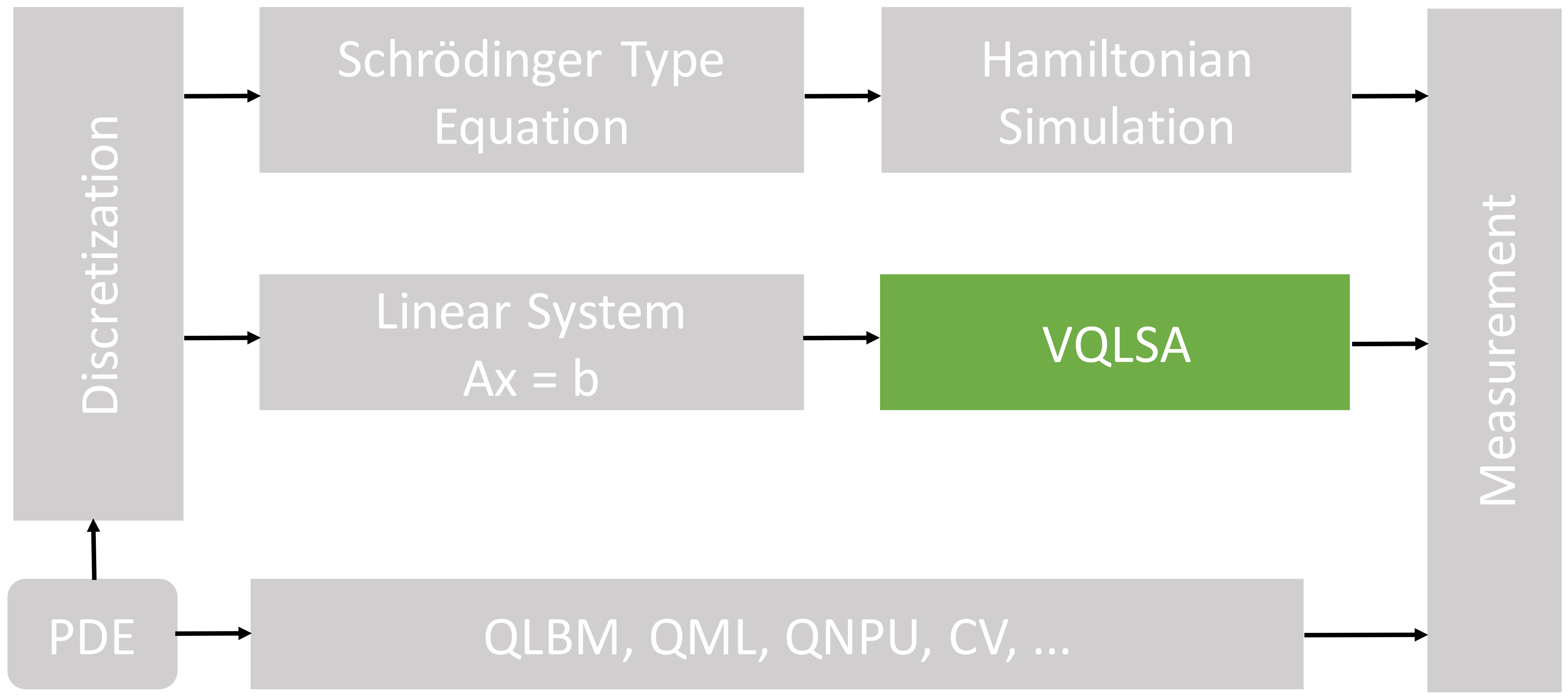
Timeline

- Session 1 (10-11:30 am)
    - Introduction (20 min)
    - Classical Solution (10 min)
    - Quantum Solution (30 min)
    - Notebook on Hamiltonian Simulation (30 min)
- Session 2 (1-2:30 pm)
    - Near-term QC (30 min)
    - Notebook on VQA (40 min)
    - Open discussion (20 min)

Timeline

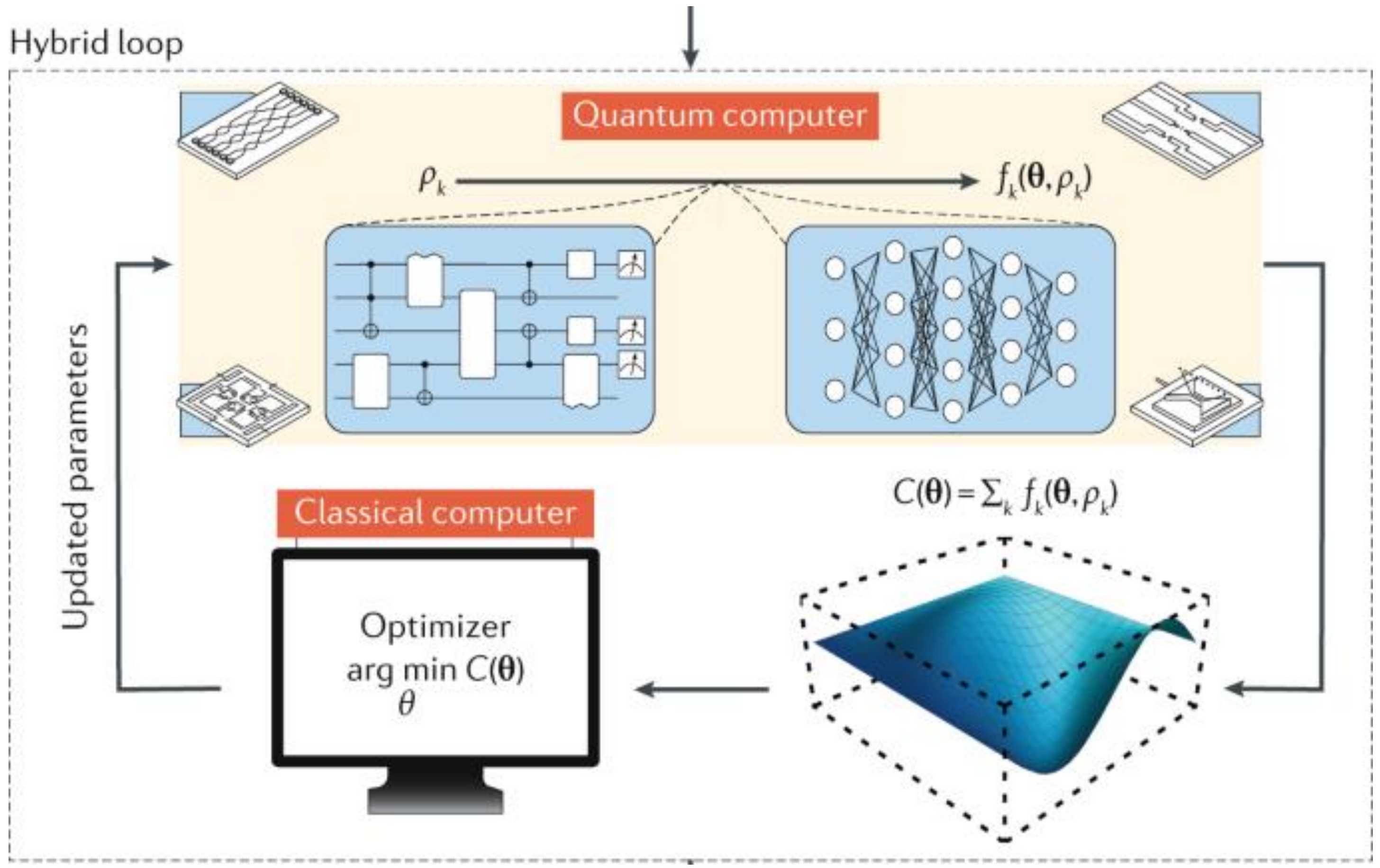
- Session 1 (10-11:30 am)
  - Introduction (20 min)
  - Classical Solution (10 min)
  - Quantum Solution (30 min)
  - Notebook on Hamiltonian Simulation (30 min)
- Session 2 (1-2:30 pm)
  - Near-term QC (30 min)
  - Notebook on VQA (40 min)
  - Open discussion (20 min)

Quantum Algorithms for solving PDEs



Near-term QC

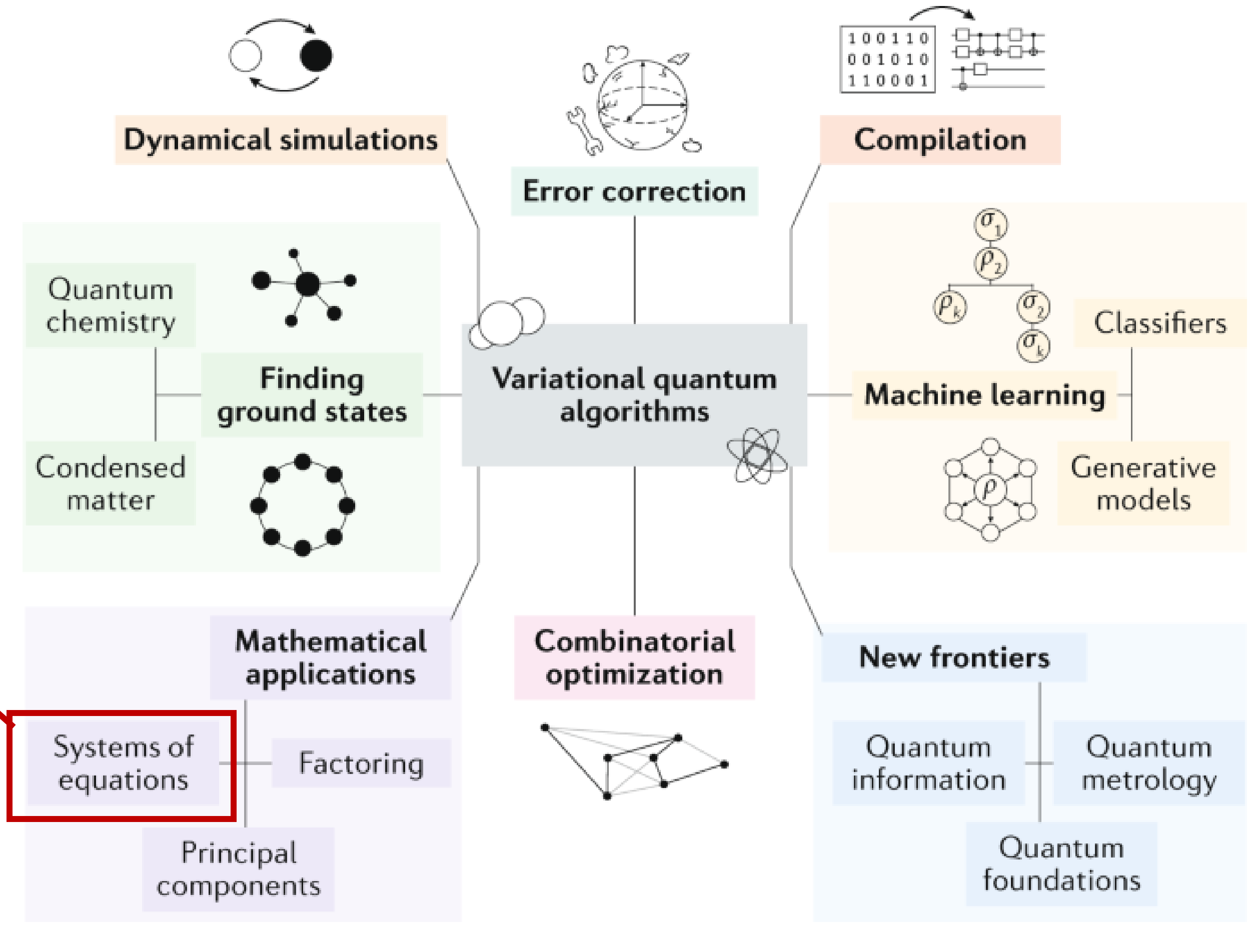
In most cases, people are referring to some Variational Quantum Algorithms (VQAs)


Hybrid loop
Quantum computer
ρ_k
$f_k(\boldsymbol{\theta}, \rho_k)$
Updated parameters
Classical computer
Optimizer
arg min C(θ)
θ
$C(\boldsymbol{\theta}) = \sum_k f_k(\boldsymbol{\theta}, \rho_k)$

Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S. C., Endo, S., Fujii, K., ... & Coles, P. J. (2021). Variational quantum algorithms. *Nature Reviews Physics*, *3*(9), 625-644.

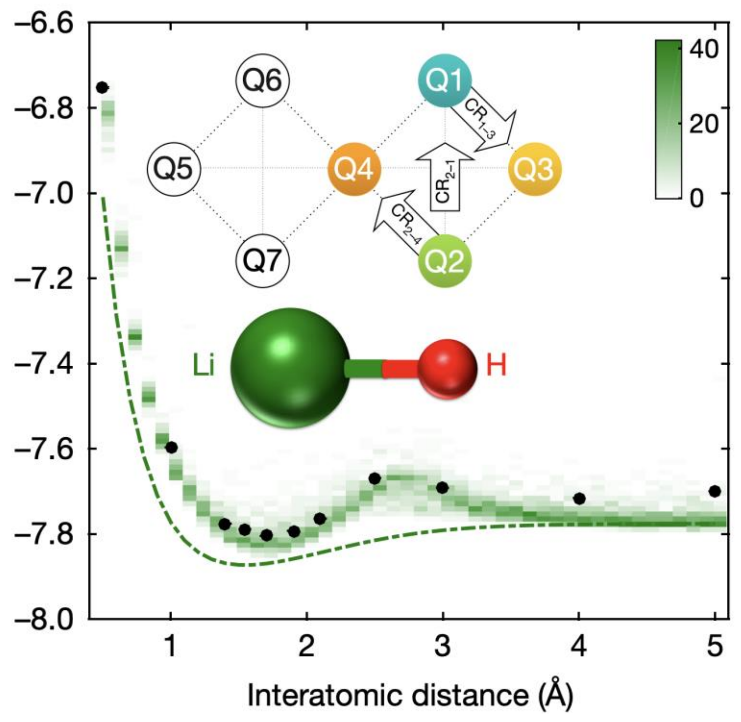
Variational quantum algorithms

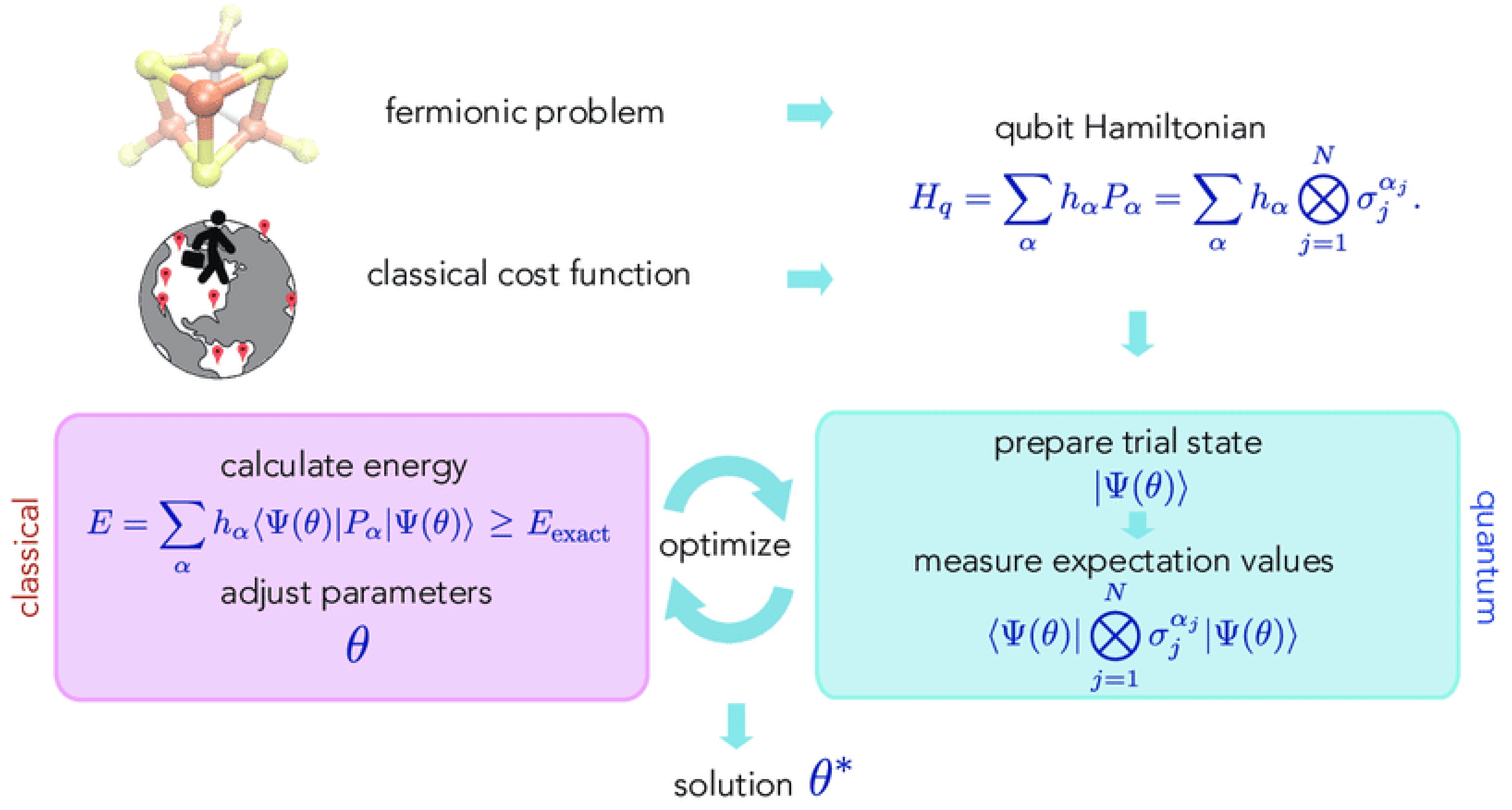
What we want!



Variational Quantum Eigensolver

VQE adopts the variational principle to estimate the ground state energy of molecules





fermionic problem
classical cost function
qubit Hamiltonian
$H_q = \sum_\alpha h_\alpha P_\alpha = \sum_\alpha h_\alpha \bigotimes_{j=1}^{N} \sigma_j^{\alpha_j}.$
classical
calculate energy
$E = \sum_\alpha h_\alpha \langle \Psi(\theta) | P_\alpha | \Psi(\theta) \rangle \geq E_{\text{exact}}$
adjust parameters
θ
optimize
prepare trial state
$|\Psi(\theta)\rangle$
measure expectation values
$\langle \Psi(\theta) | \bigotimes_{j=1}^{N} \sigma_j^{\alpha_j} | \Psi(\theta) \rangle$
quantum
solution θ^*

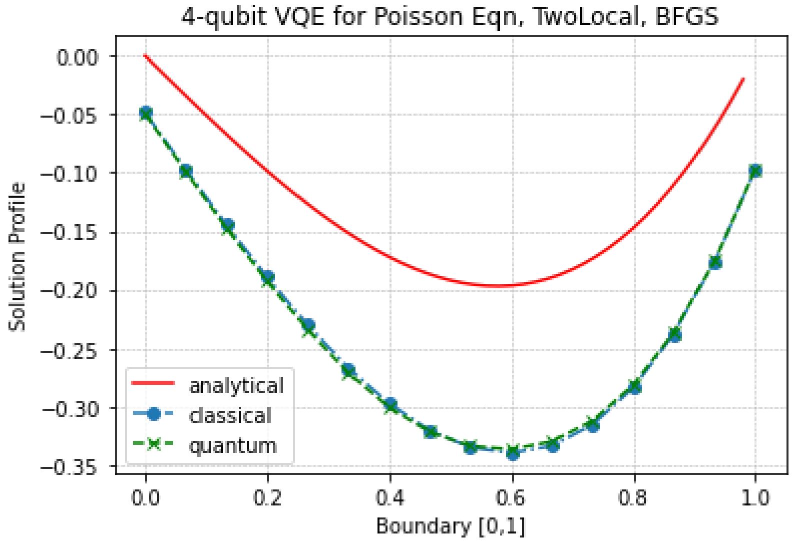
Moll, N., Barkoutsos, P., Bishop, L. S., Chow, J. M., Cross, A., Egger, D. J., ... & Temme, K. (2018). Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Science and Technology*, *3*(3), 030503.

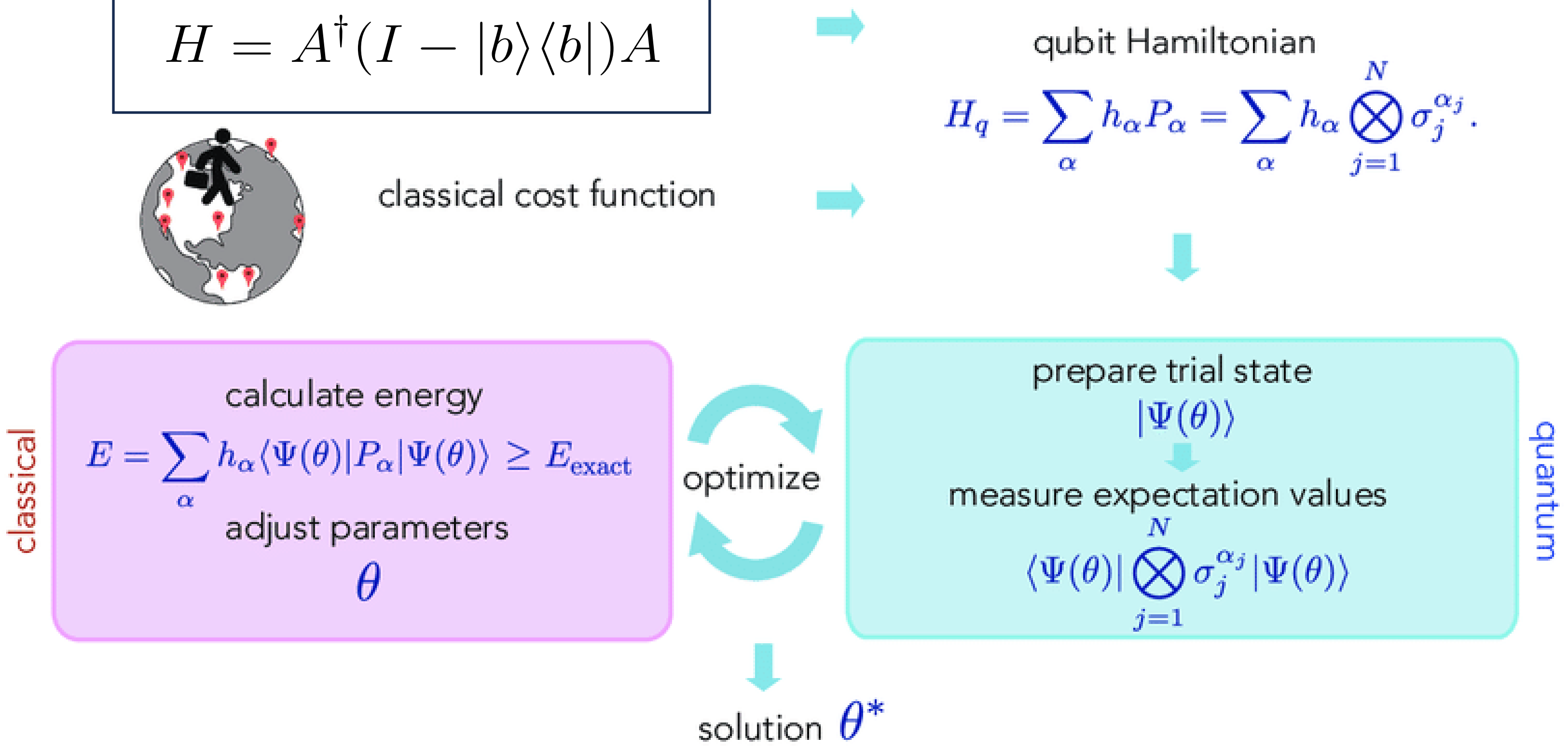
VQE as VQLSA

Encode the solution of linear system into the ground state of a target Hamiltonian

Fake problem

$$H = A^\dagger (I - |b\rangle\langle b|) A$$




classical cost function
qubit Hamiltonian
$H_q = \sum_\alpha h_\alpha P_\alpha = \sum_\alpha h_\alpha \bigotimes_{j=1}^{N} \sigma_j^{\alpha_j}.$
calculate energy
$E = \sum_\alpha h_\alpha \langle\Psi(\theta)|P_\alpha|\Psi(\theta)\rangle \geq E_{\text{exact}}$
adjust parameters
θ
classical
optimize
prepare trial state
$|\Psi(\theta)\rangle$
measure expectation values
$\langle\Psi(\theta)|\bigotimes_{j=1}^{N} \sigma_j^{\alpha_j}|\Psi(\theta)\rangle$
quantum
solution θ^*

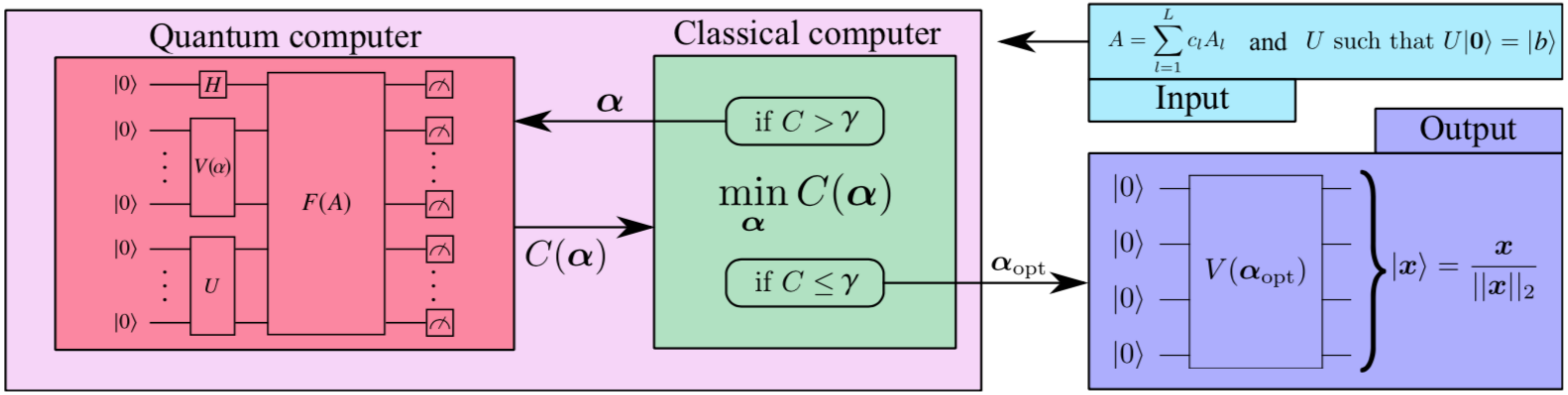
Moll, N., Barkoutsos, P., Bishop, L. S., Chow, J. M., Cross, A., Egger, D. J., ... & Temme, K. (2018). Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Science and Technology*, *3*(3), 030503.

Variational Quantum Linear Solver

Similar idea but different cost function:

$$\left.\begin{array}{lcll} |\Phi\rangle \perp |b\rangle & \Rightarrow & C(\Theta) & \text{large} \\ |\Phi\rangle // \ |b\rangle & \Rightarrow & C(\Theta) & \text{small} \end{array}\right\} \quad |\Phi\rangle = A|\psi(\Theta)\rangle$$

$$\hat{C}(\Theta) = 1 - \frac{|\langle \Phi \mid b \rangle|^2}{\langle \Phi \mid \Phi \rangle}$$

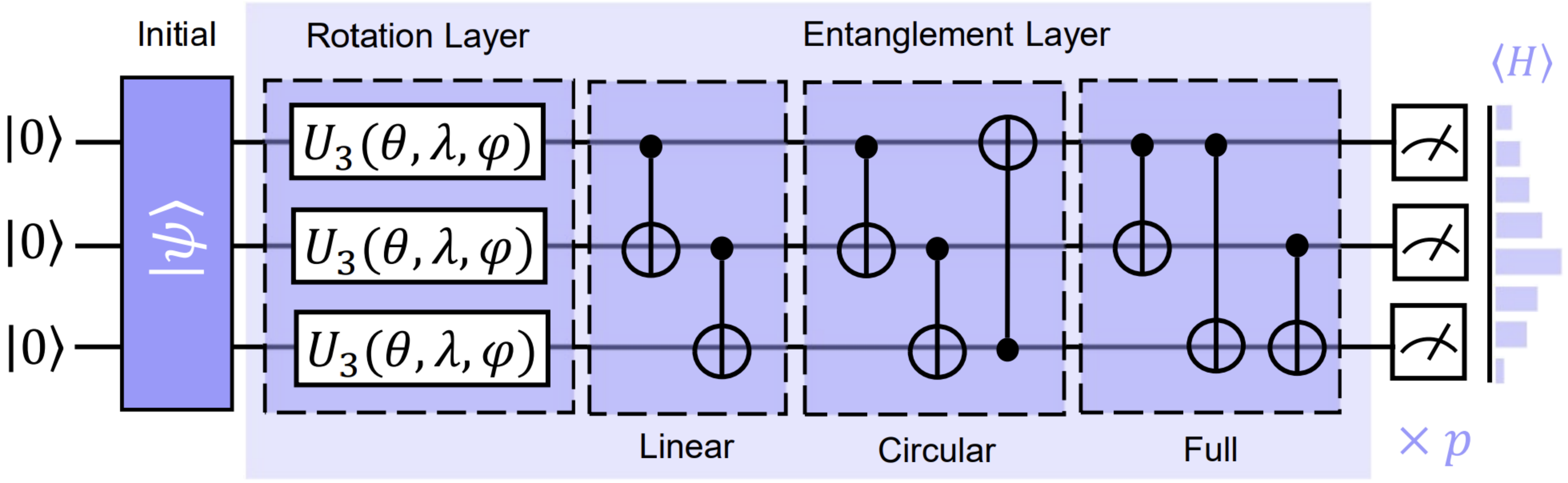


Quantum computer
Classical computer
$A = \sum_{l=1}^{L} c_l A_l$ and U such that $U|\mathbf{0}\rangle = |b\rangle$
Input
Output
$|0\rangle$
H
$V(\alpha)$
U
$F(A)$
$\boldsymbol{\alpha}$
$C(\boldsymbol{\alpha})$
if $C > \gamma$
$\min_{\boldsymbol{\alpha}} C(\boldsymbol{\alpha})$
if $C \leq \gamma$
$\boldsymbol{\alpha}_{\text{opt}}$
$V(\boldsymbol{\alpha}_{\text{opt}})$
$|\boldsymbol{x}\rangle = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2}$

Bravo-Prieto, C., LaRose, R., Cerezo, M., Subasi, Y., Cincio, L., & Coles, P. J. (2019). Variational quantum linear solver. *arXiv preprint arXiv:1909.05820*.

Ansatz

- Find optimal problem-specific circuit structure is difficult due to large search space
- Hardware-Efficient Ansatz (HEA) is usually the choice if we know nothing about the problem structure or symmetry



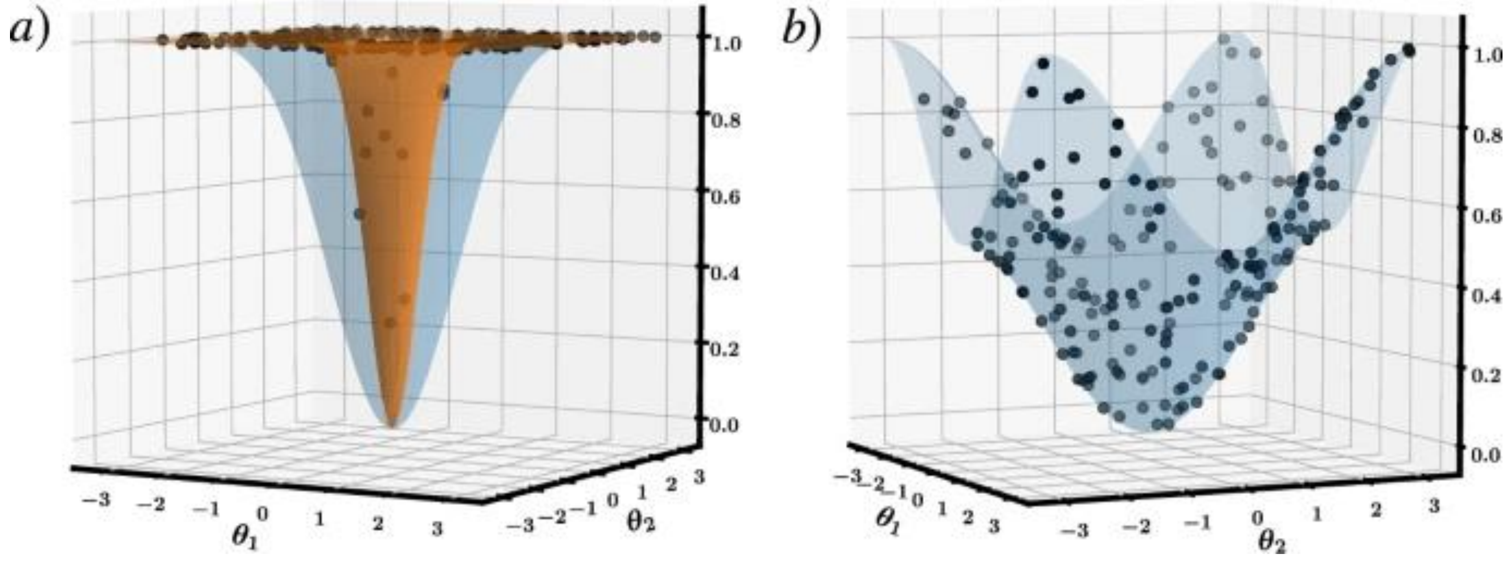
Optimizer

- Gradient based (converge faster)
  - Gradient Descent (GD)
  - SGD
  - Adam
  - BFGS (my favorite)
- Non-gradient based (more robust to noise)
  - COBYLA
  - Nelder-Mead
  - Powell
  - SPSA

There is no best choice. You need to test various things.

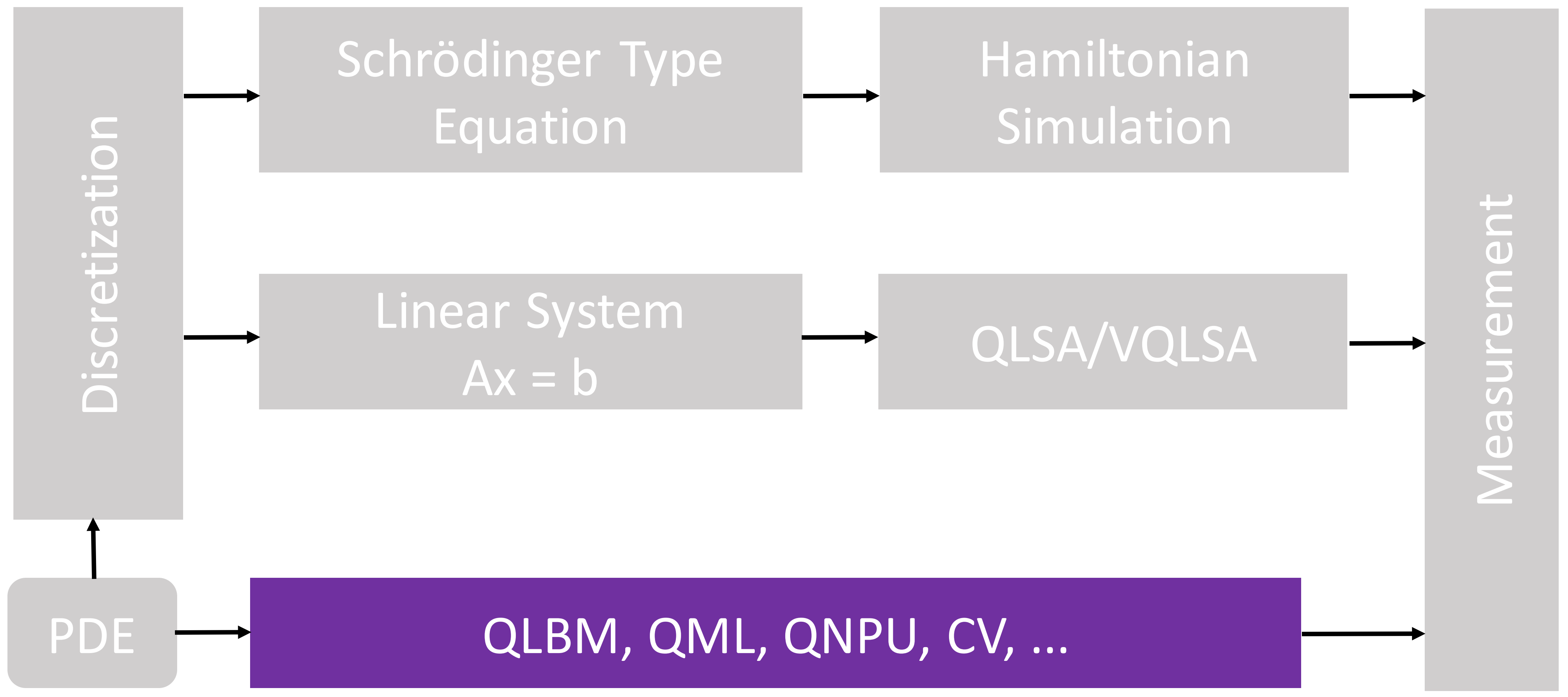
Barren Plateau (BP)

- When a given cost function exhibits a BP, the magnitude of its partial derivatives will be, on average, exponentially vanishing with the system size
- This makes VQE+HEA training difficult for large qubits

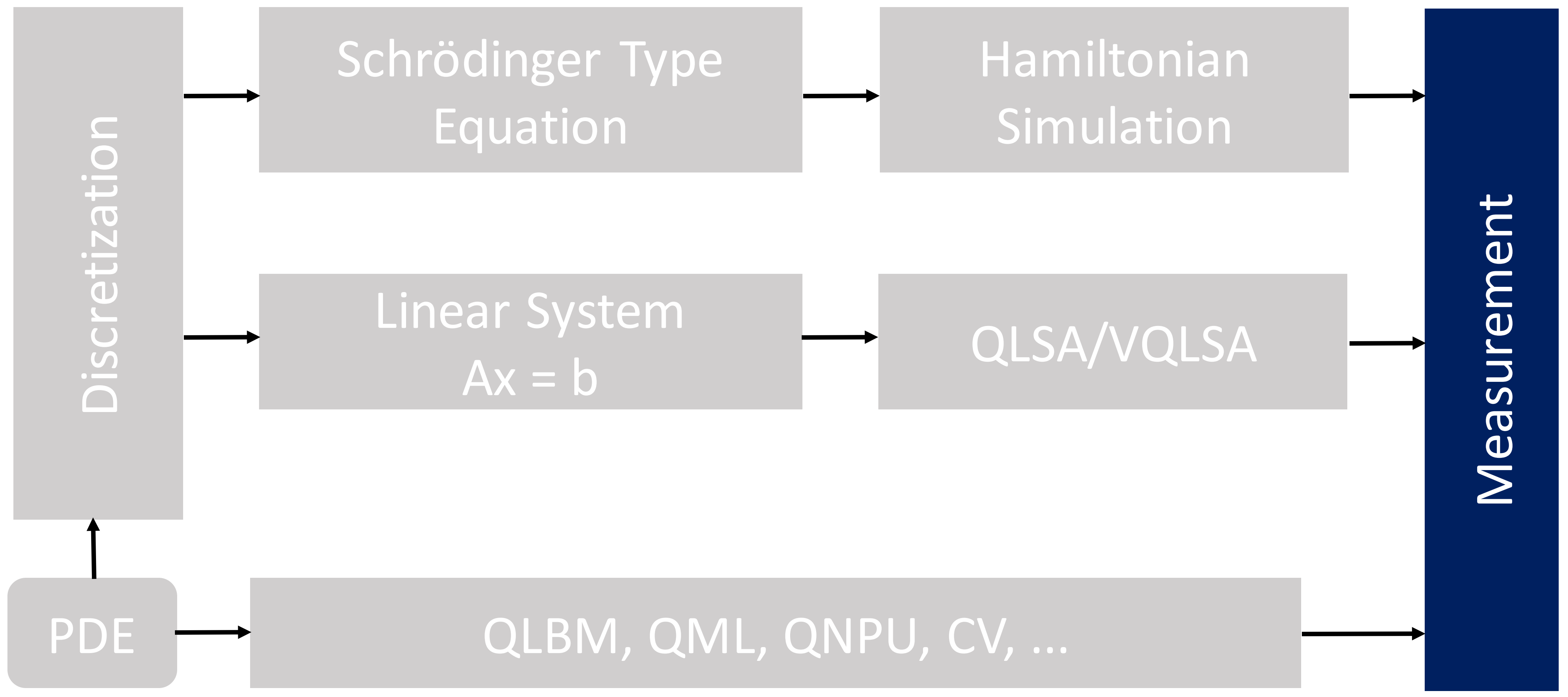


McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R., & Neven, H. (2018). Barren plateaus in quantum neural network training landscapes. *Nature communications*, *9*(1), 4812.
Cerezo, M., Sone, A., Volkoff, T., Cincio, L., & Coles, P. J. (2021). Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature communications*, *12*(1), 1791.

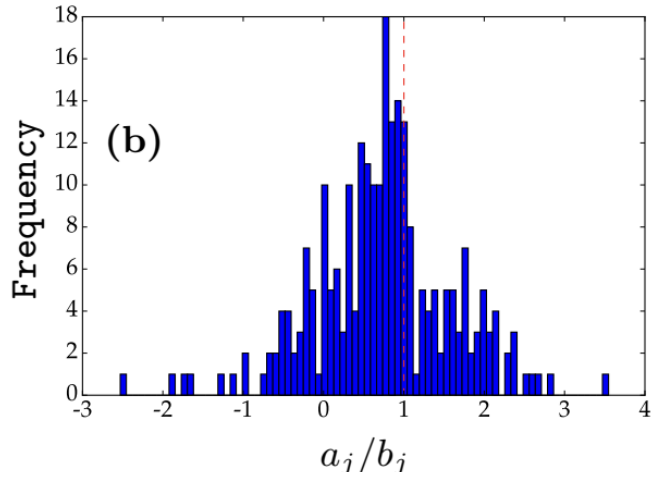
Quantum Algorithms for solving PDEs



Quantum Algorithms for solving PDEs



Fidelity Estimation



- Expand two states in the Pauli basis as

$$\rho = \sum_j a_j W_j / 2^{N/2} \qquad \psi = \sum_j b_j W_j / 2^{N/2}$$

- The fidelity can be estimated as $F(\psi, \rho) = \operatorname{tr}(\psi\rho) = \sum_j a_j b_j = \sum_j \left(\frac{a_j}{b_j}\right) b_j^2$
- Notice $a_j/b_j = \operatorname{tr}(W_j\rho) / \operatorname{tr}(W_j\psi)$
- Then we can sample these observables to obtain a randomized estimation of fidelity with sample $O\left(2^n/\epsilon^2\right)$

Take-home message

- Steady state problems are more suitable to solve for near-term
- Quantum algorithms are not good at nonlinearity
- Explore more near-term algorithms for linear system
- Possible research direction is simulating QLSA such as HHL and improved version in HPC systems

Timeline

- Session 1 (10-11:30 am)
  - Introduction (20 min)
  - Classical Solution (10 min)
  - Quantum Solution (30 min)
  - Notebook on Hamiltonian Simulation (30 min)
- Session 2 (1-2:30 pm)
  - Near-term QC (30 min)
  - Notebook on VQA (40 min)
  - Open discussion (20 min)

Thanks for your attention!

- Q&A
- Coding time!
- We are looking for collaborations if you are interested.

Timeline

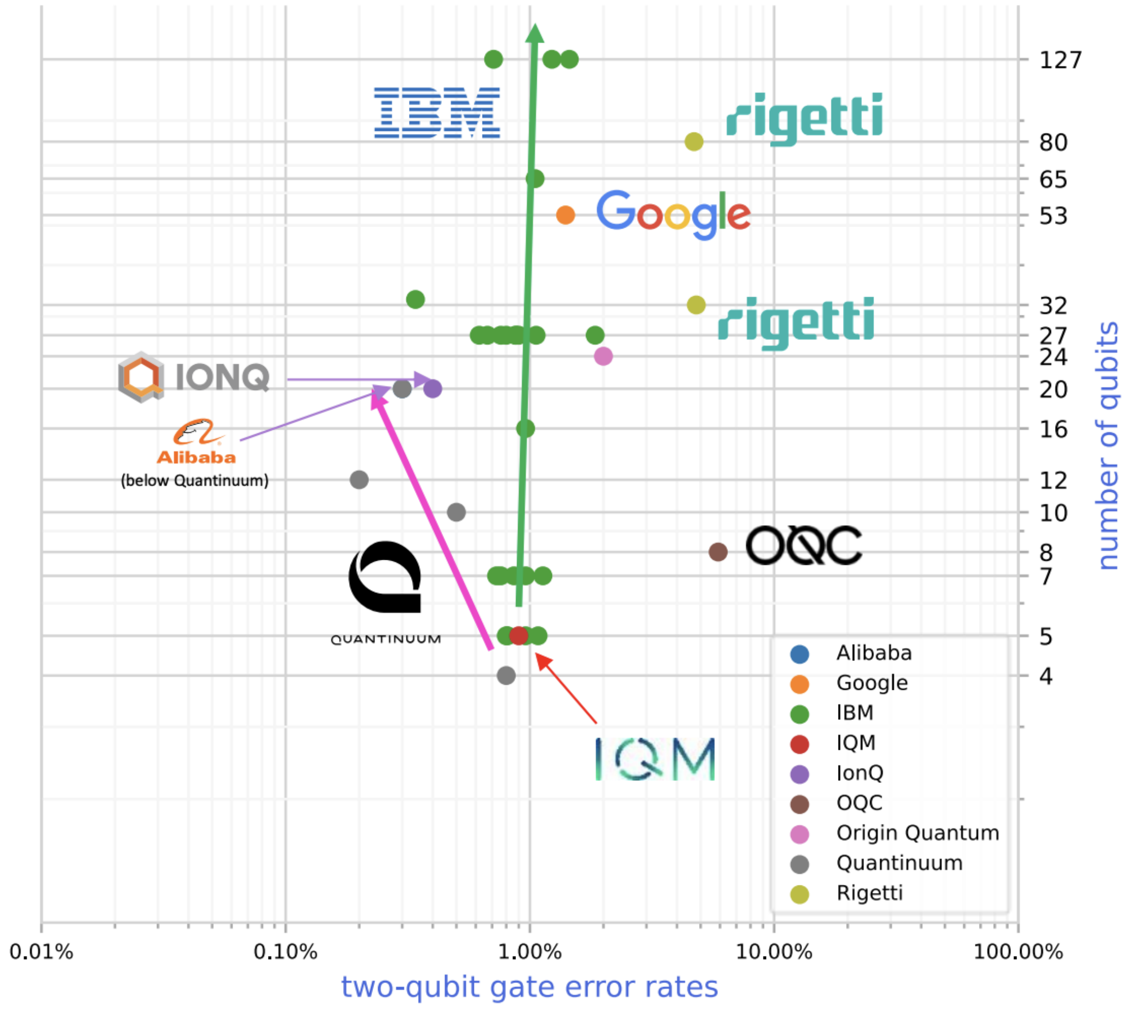
- Session 1 (10-11:30 am)
    - Introduction (20 min)
    - Classical Solution (10 min)
    - Quantum Solution (30 min)
    - Notebook on Hamiltonian Simulation (30 min)
- Session 2 (1-2:30 pm)
    - Near-term QC (30 min)
    - Notebook on VQA (40 min)
    - Open discussion (20 min)

Software

- QuDiffEq.jl https://github.com/QuantumBFS/QuDiffEq.jl
- SimuQ https://pickspeng.github.io/SimuQ/

How to choose your backend?

- Through word of mouth, many people are in favor of Quantinuum's hardware due to high fidelity.
- What's your experience on this?
- Have you considered to run your algorithm on the actual quantum hardware?



Ezratty, O. (2023). Is there a Moore's law for quantum computing?. *arXiv preprint arXiv:2303.15547*.

How to access backend?

- OLCF
- AWS Research Credit
- Microsoft Azure Credit
- IonQ Research Proposal
- IBM Quantum free/paid plans