OAK RIDGE National Laboratory | LEADERSHIP COMPUTING FACILITY

August 6, 2025

Evaluating Quantum Linear Solvers for Fluid Flow Enabled Through OLCF’s QCUP

2025 OLCF User Meeting

Murali Gopalakrishnan Meena
Oak Ridge National Laboratory

U.S. DEPARTMENT OF ENERGY

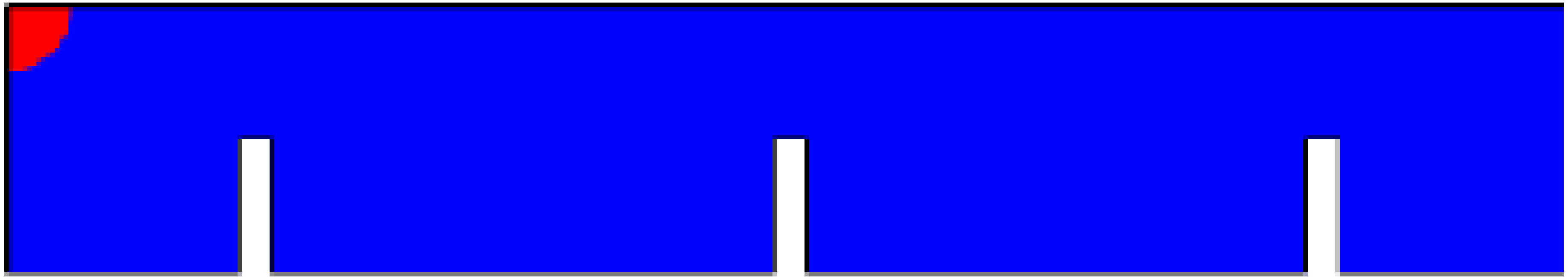
ORNL IS MANAGED BY UT-BATTELLE LLC
FOR THE US DEPARTMENT OF ENERGY

*This research used resources of the **Oak Ridge Leadership Computing Facility (OLCF) at the Oak Ridge National Laboratory**, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.*

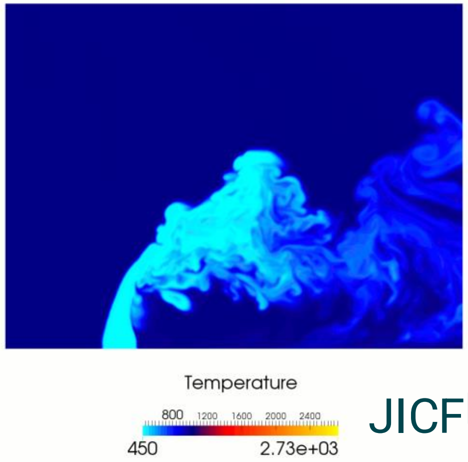
Introduction & Motivation

Fluid dynamics & Extreme-scale computation

Fluid dynamics is everywhere & its “pretty” turbulent

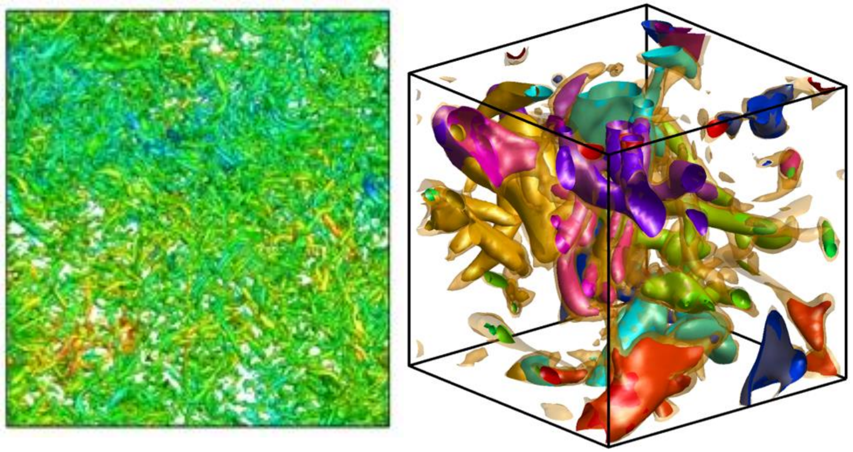


Deflagration to Detonation Transition in fuel-air mixtures accelerated by obstacles[1]



JICF[2]

> I am an old man now, and when I die and go to Heaven there are two matters on which I hope for enlightenment. One is **quantum electrodynamics** and the other is the **turbulent motion of fluids**. And **about the former I am rather more optimistic.**
>
> - Sir Horace Lamb



Community structures in 3D isotropic turbulence[3]

[1] Gottiparthi and Menon, Proc. European Combust. Meeting, 2013
[2] Gottiparthi et al., AIAA-2016-4791, 2016
[3] Gopalakrishnan Meena & Taira, *J. Fluid Mech.*, 2021

Governing equations of fluid flow comprise of linear & nonlinear terms that can be solved numerically

The Navier–Stokes Equations

Conservation of mass

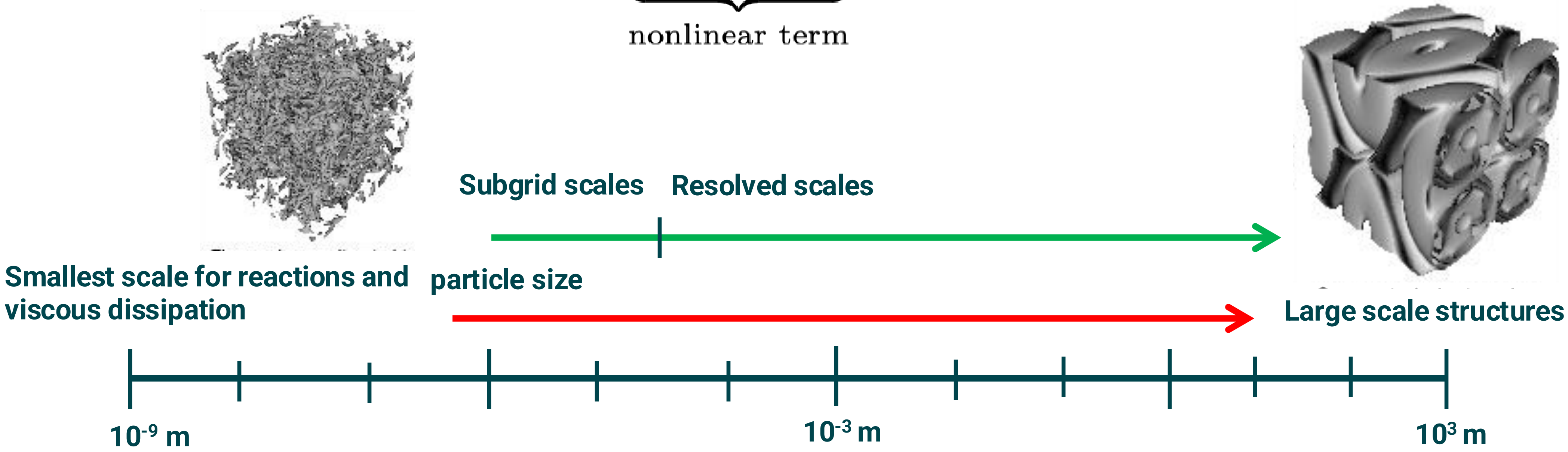
$$\nabla \cdot \boldsymbol{u} = 0,$$

Conservation of momentum

$$\frac{\partial \boldsymbol{u}}{\partial t} = -\underbrace{\boldsymbol{u} \cdot \nabla \boldsymbol{u}}_{\text{nonlinear term}} \underbrace{- \nabla p + \frac{1}{Re} \nabla^2 \boldsymbol{u}}_{\text{linear terms}} + \boldsymbol{s}$$

Pressure Poisson equation

$$\nabla^2 p = -\underbrace{\nabla \cdot (\boldsymbol{u} \cdot \nabla \boldsymbol{u})}_{\text{nonlinear term}}$$



Subgrid scales

Resolved scales

Smallest scale for reactions and viscous dissipation

particle size

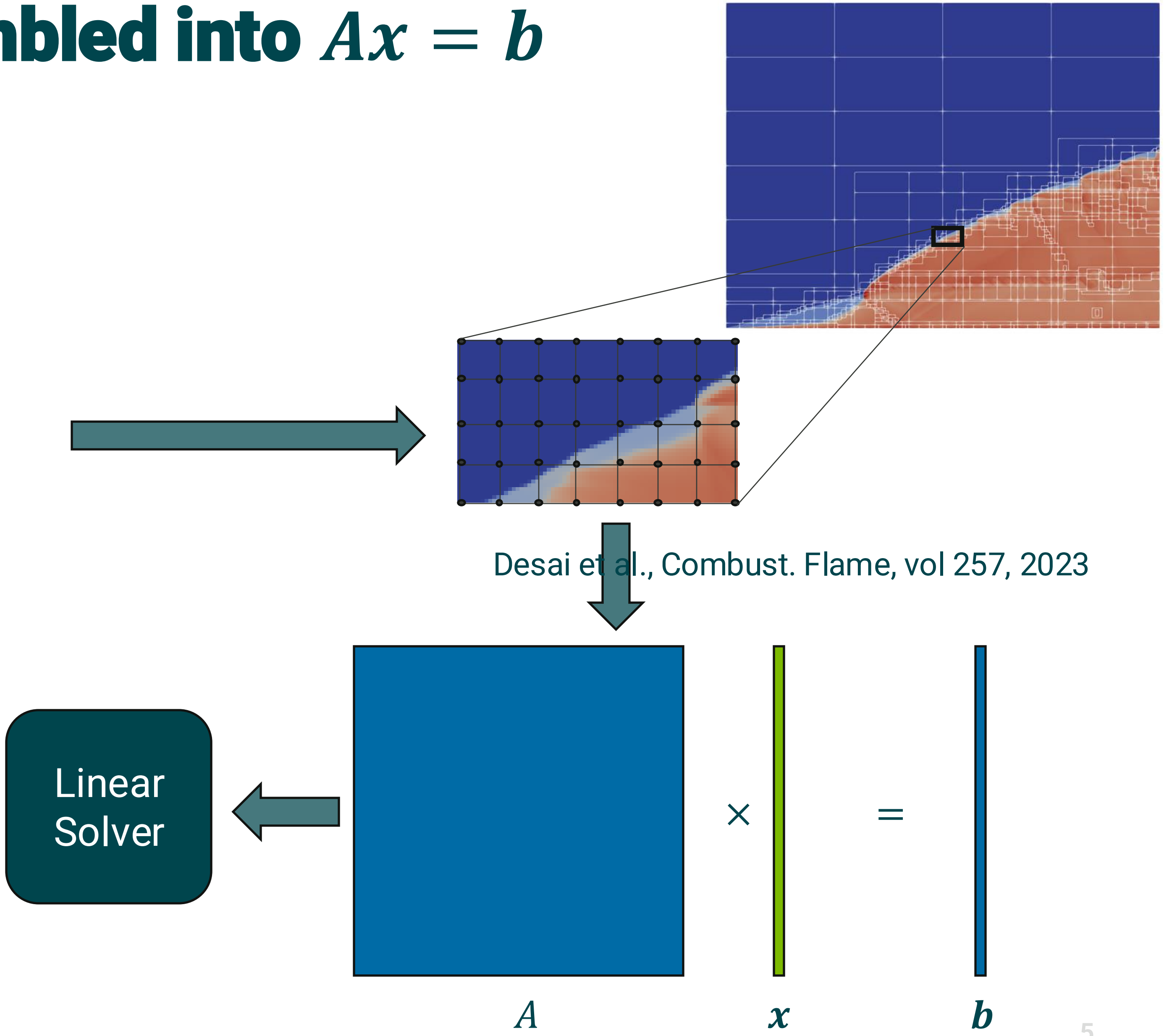
Large scale structures

Governing equations are discretized to create a set of algebraic equations & assembled into $Ax = b$

$$\nabla \cdot \boldsymbol{u} = 0,$$

$$\frac{\partial \boldsymbol{u}}{\partial t} = -\underbrace{\boldsymbol{u}\cdot\nabla\boldsymbol{u}}_{\text{nonlinear term}} \underbrace{-\nabla p + \frac{1}{Re}\nabla^2\boldsymbol{u}}_{\text{linear terms}} + \boldsymbol{s}$$

$$\nabla^2 p = -\underbrace{\nabla\cdot(\boldsymbol{u}\cdot\nabla\boldsymbol{u})}_{\text{nonlinear term}}$$

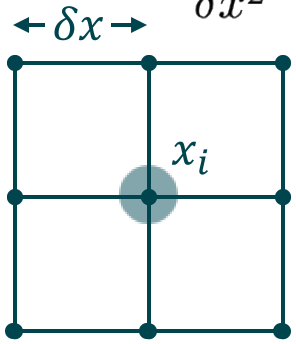


Desai et al., Combust. Flame, vol 257, 2023

Finite difference approximation

$$\frac{\partial u_\alpha(x_i)}{\partial x} \simeq \frac{u_\alpha(x_i+\delta x) - u_\alpha(x_i-\delta x)}{2\delta x}$$

$$\frac{\partial^2 u_\alpha(x_i)}{\partial x^2} \simeq \frac{u_\alpha(x_i+\delta x) + u_\alpha(x_i-\delta x) - 2u_\alpha(x_i)}{\delta x^2}$$

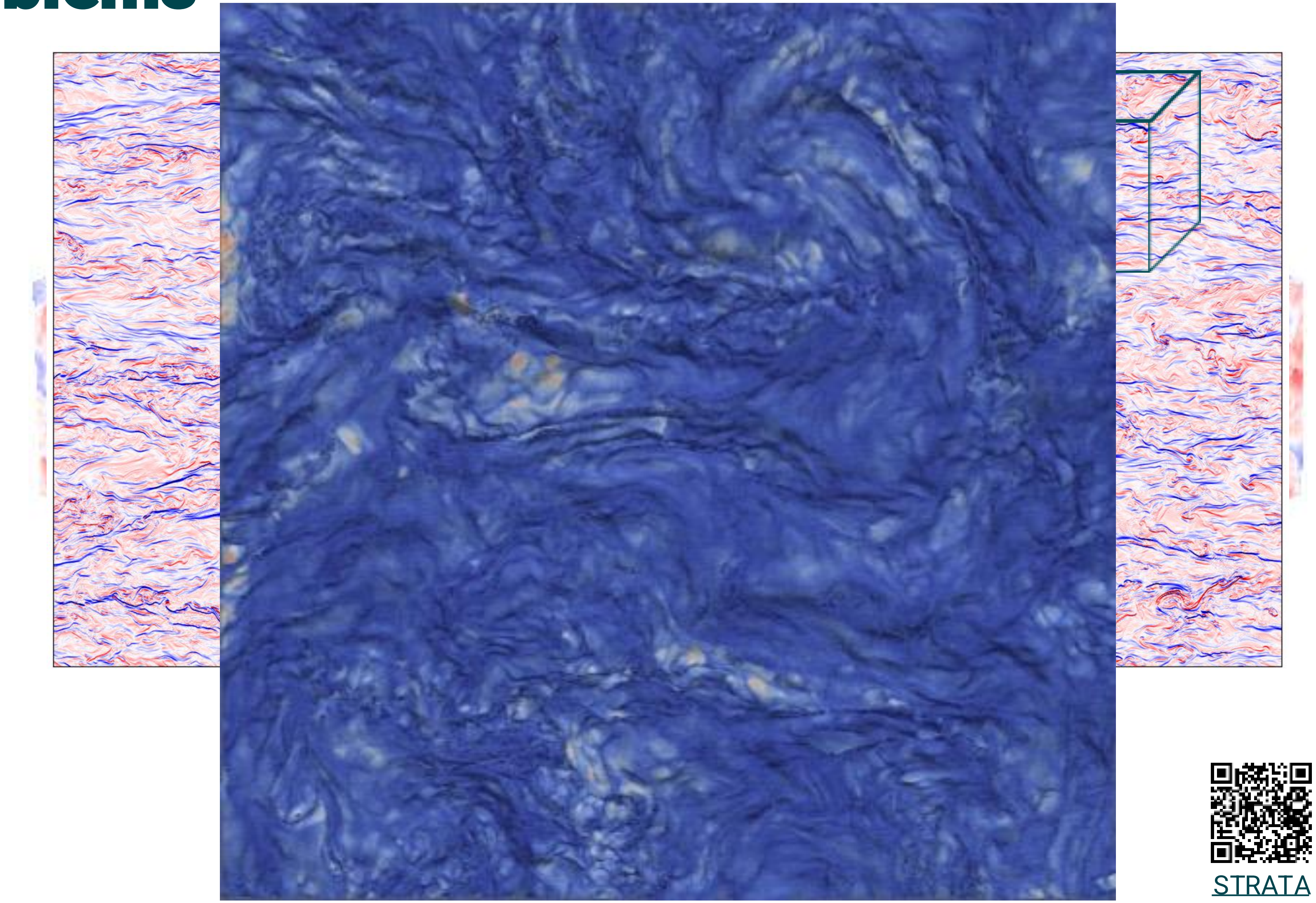


Grid resolution can get prohibitively expensive to simulate for practical fluid flow problems

- 3D stratified turbulence model for oceanographic flow[1]

- $37\text{k} \times 37\text{k} \times 4\text{k} \approx 6.6 \times 10^{12} > 2^{42}$ grid points

- 100 TB per snapshot



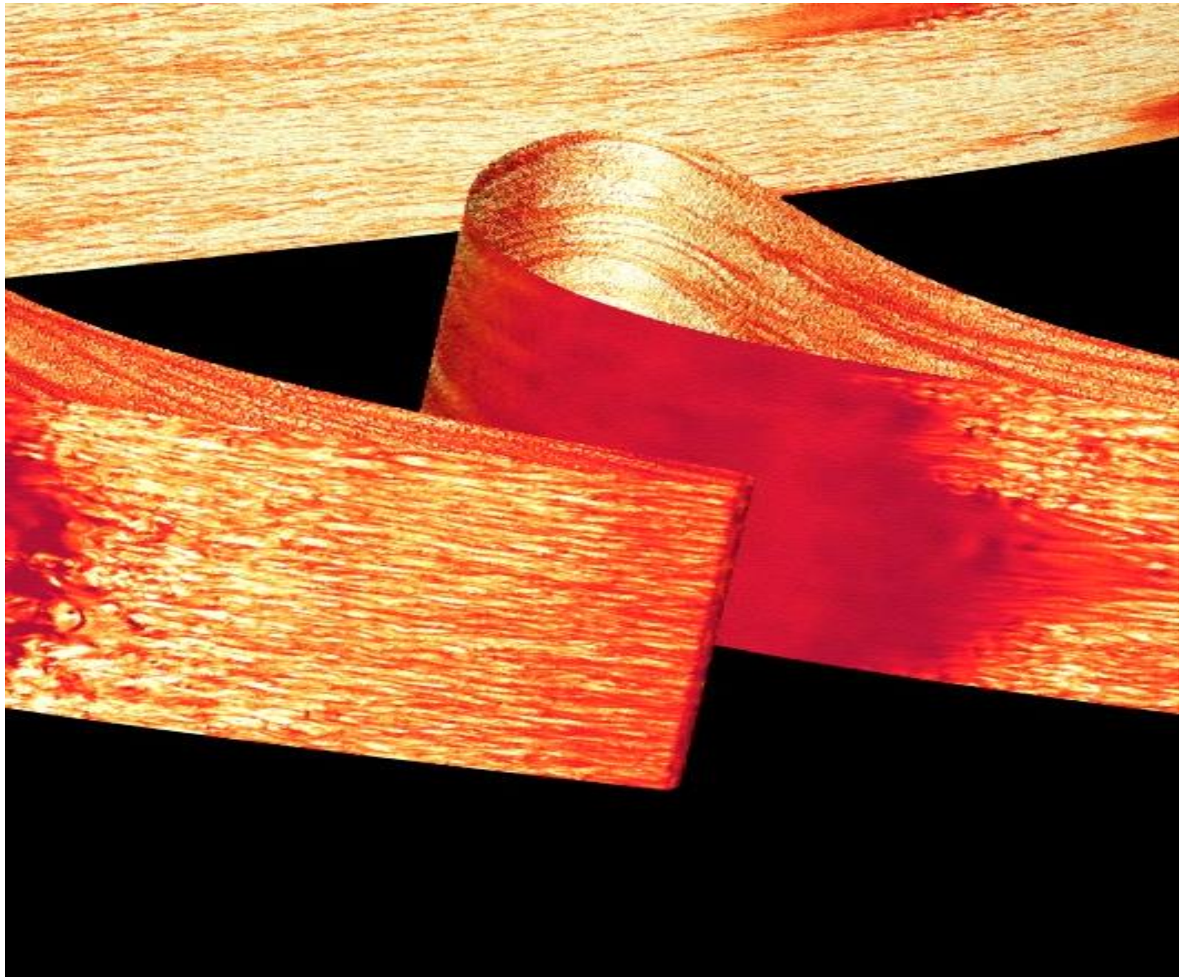
Isosurface of scalar dissipation rate for 1/25th of the domain, constructed using 127×10^6 triangles.

[1] J. J. Riley, et al., *J. Turbulence*, 2023

STRATA Group

Grid resolution can get prohibitively expensive to simulate for practical fluid flow problems

- Flow past turbine blades showing instantaneous heat transfer[1]
- Transition to turbulence is very challenging to capture
- Turbulence related projects: 35-45% of 2023 OLCF Frontier allocation



Isosurface of heat flux simulated using 14.6 billion cells.

[1] T. Jelly et al., *J. Turbomach.*, 2025

Quantum linear & nonlinear PDE solvers have the potential to exponentially reduce cost of solving large problems

- Quantum computing applications to fluid flow problems:
  - Lattice simulations: fluid motion modeled as the motion of discrete particles
  - **Continuum simulations**: fluid motion modeled as a continuous field

Linear flow problems

- Linear (ideal) & Linearized PDEs[1-4]: N—S equations with assumptions
- Use **Quantum Linear System Algorithms (QLSA)**
- Classical: $\boldsymbol{O(N)}$ (or higher for denser non-symmetric matrices)
- Quantum: $\boldsymbol{O(log(N))}$
- Disadvantage: Enlarged solution space

[1] Yepez, *PRE*, 2001
[2] Xu, Daley, Givi, Somma, *AIAAJ*, 2018
[3] Bharadwaj & Sreenivasan, *PNAS*, 2023
[4] Gopalakrishnan Meena at al., *PoF*, 2024

Nonlinear flow problems

- Tackle nonlinearity of PDEs[5-10]
- Not generalized
- Limited work
- Variational algorithm using **Quantum Nonlinear Processing Units**[8]

[5] Leyton & Osborne, *arXiv*, 2008
[6] Gaitan, *NPJ*, 2020
[7] Steijl, *Quantum Comp. & Comm.*, 2020
[8] Oz, et al., *Sci. Rep.*, 2023
[9] Lubasch, et al., *PRA*, 2020
[10] Gopalakrishnan Meena et al., *IEEE QCE*, 2024

Objectives of this presentation

- Disclaimer: We are **not** trying to show/demonstrate quantum advantage for fluid flow problems
- Current talk objectives:
  - Efforts at OLCF to investigate the **application of a QLSA on a canonical fluid flow problem**
  - Focus on **practical issues** on using the algorithm for canonical fluid flow problems
    - Computational cost
    - Noise modeling & mitigation
    - Running on real hardware
  - **Collaborative effort** & welcome collaboration

ORNL QCFD team and collaborators

Group website

Kalyan
Gottiparthi

Chao
Lu

Toño
Coello Pérez

Amir
Shehata

Michael
Sandoval

Seongmin
Kim

In-Saeng
Suh

Antigoni
Georgiadou

Alessandro
Baroni

Ryan
Landfield

Matt
Norman

Tom
Beck

ORNL - OLCF

Pooja
Rao
NVIDIA

Justin
Lietz
NVIDIA

Xinfeng
Gao
UVA

Paul
Lin
NERSC-LBNL

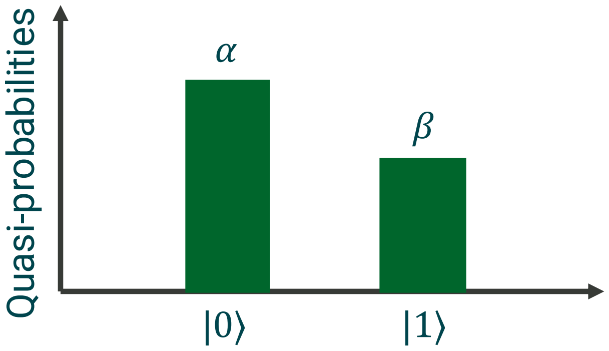
Yu
Zhang
LANL

Quantum Computing

A very brief Intro to QC

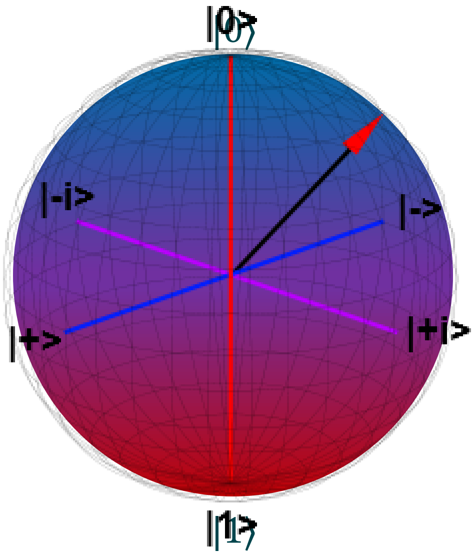
Classical vs Quantum Computing

- Bit vs Qubit – information states
- State of superposition – a combination of all possible configurations
  - Store more information!
  - Measuring will lead to collapse to a binary state
- Types of qubits:
  - Superconducting
  - Trapped ion
  - Photons
  - Neutral atoms
  - Quantum dots



Classical computing - **Bit**

Quantum computing - **Qubit**



$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

https://www.ibm.com/think/topics/quantum-computing
Rick Arthur's Quantum Computing 101 briefing: https://drive.google.com/file/d/1B-V1lYjKFUXCVIZybcjFFeqW-fZUUHFE/view?usp=sharing

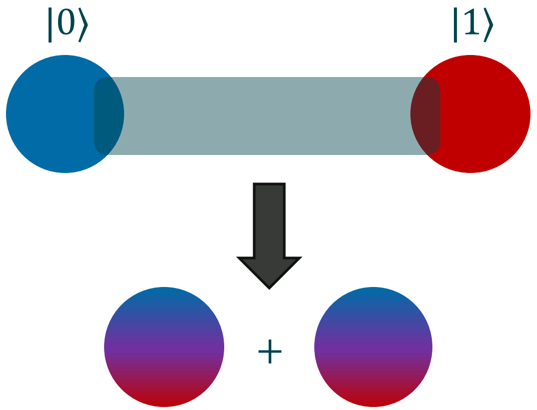
Key principles of quantum computing

1. **Superposition**

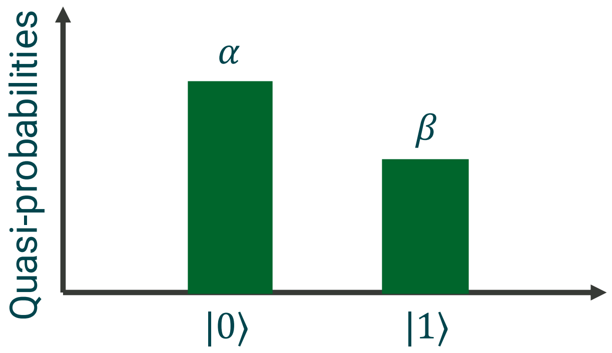
2. **Entanglement** – ability of qubits to correlate with each others' states
   - Store even more information!

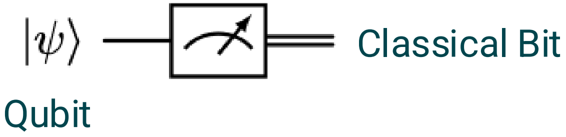
3. **Interference** – information is structured like waves with amplitudes
   - Waves can amplify or cancel each other
   - Amplitudes: Probabilities of the outcomes of measurement

4. **Decoherence** – collapse from quantum to nonquantum state
   - Intentionally (measurement)
   - Allows quantum computers to interact with classical computers
   - Unintentionally (interaction with environment)



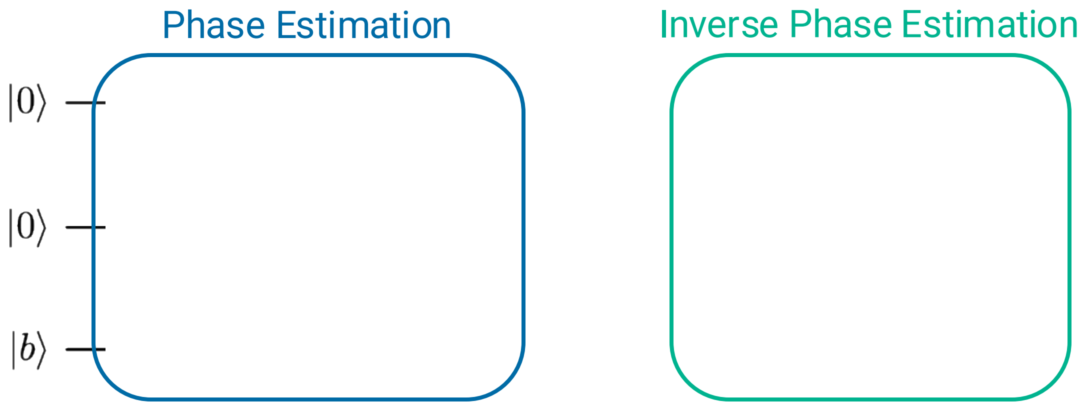
$$|\psi\rangle = \alpha_1|00\rangle + \alpha_2|01\rangle + \alpha_3|10\rangle + \alpha_4|11\rangle$$





Working of a quantum computer

1. **Qubits** are prepared as superposition of states
2. **Gates** used to operate on qubits and entangle them
   - Unitary operations
   - Reversible
   - Can operate on a single qubit or multiple (entanglement)
3. **Circuits** are collection of gates
4. **Quantum algorithms** are collection of circuits to create desired interference between states
5. **Measurement** (amplified outcomes) gives solution



Running a quantum algorithm

Backend	Mechanism	Functionality
Simulator	Classical	Classical program modeling a quantum system in an ideal scenario
Emulator	Classical	Classical program modeling actual behavior of a quantum system
Real	Quantum	Physical hardware performing real quantum computations

Quantum Linear Solver Algorithms

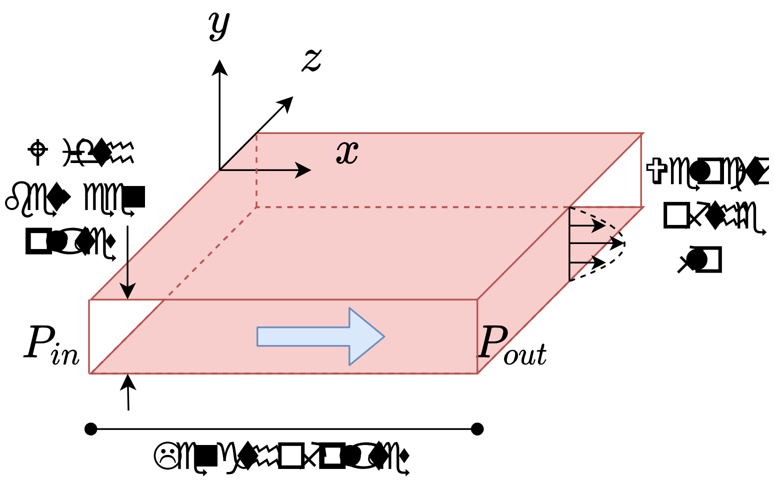
Solving the Hele–Shaw flow using QLSA

Model problem: Hele–Shaw flow

Flow between 2 flat plates driven by pressure difference at inlet & outlet

Flow properties:

- **2D, Steady**
- Incompressible, inviscid



- N–S equations reduce to

$$\nabla \cdot \boldsymbol{u} = 0 \text{ and } \Delta \boldsymbol{u} - \nabla p = 0$$

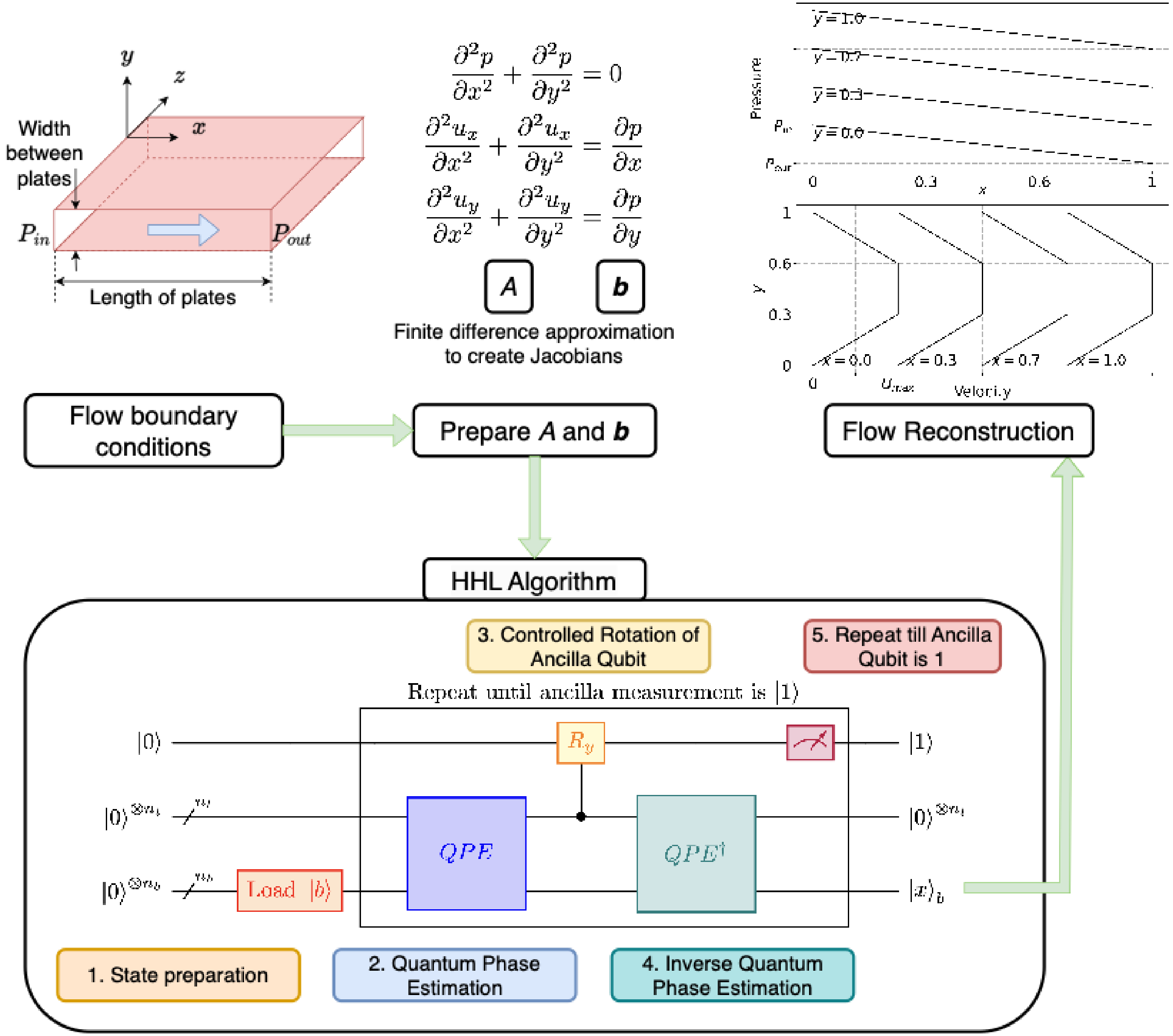
- Decoupling pressure & velocity converts the problem into the form $A\boldsymbol{x}=\boldsymbol{b}$

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = 0$$

$$\frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_x}{\partial y^2} = \frac{\partial p}{\partial x}$$

$$\frac{\partial^2 u_y}{\partial x^2} + \frac{\partial^2 u_y}{\partial y^2} = \frac{\partial p}{\partial y}$$

Overview of solving the Hele—Shaw flow using a QLSA



$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = 0$$

$$\frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_x}{\partial y^2} = \frac{\partial p}{\partial x}$$

$$\frac{\partial^2 u_y}{\partial x^2} + \frac{\partial^2 u_y}{\partial y^2} = \frac{\partial p}{\partial y}$$

Finite difference approximation to create Jacobians

Flow boundary conditions

Prepare *A* and ***b***

HHL Algorithm

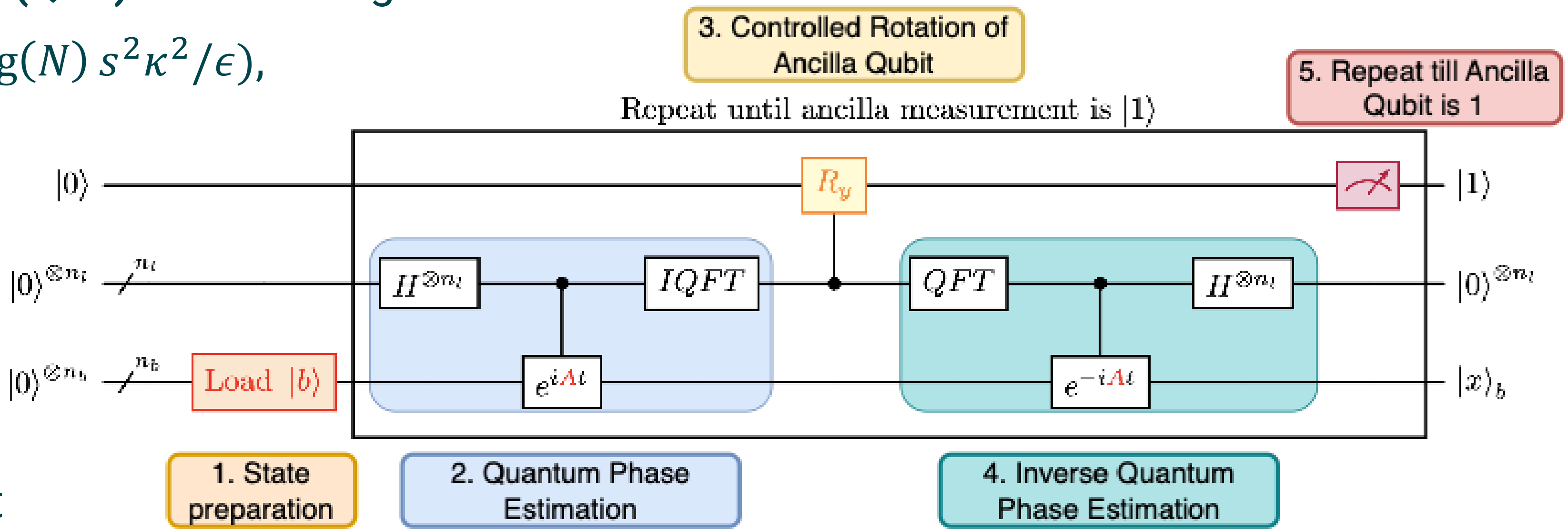
Flow Reconstruction

1. State preparation
2. Quantum Phase Estimation
3. Controlled Rotation of Ancilla Qubit
4. Inverse Quantum Phase Estimation
5. Repeat till Ancilla Qubit is 1

Repeat until ancilla measurement is $|1\rangle$

The Harrow–Hassidim–Lloyd (HHL) algorithm

- Convert the problem using eigen basis of **Hermitian** A to give
  - $\boldsymbol{x} = A^{-1}\boldsymbol{b} = \sum_j \lambda_j^{-1} b_j u_j,$ λ_j & u_j are the eigenvalues & eigenvectors of A
  - Usually, $2N$ since fluid flow Jacobians are not usually Hermitian
  - Use **Quantum Phase Estimation (QPE)** to obtain eigen basis
  - Computational complexity: $O(\log(N)\, s^2\kappa^2/\epsilon)$,
    - N – size of A
    - s – sparsity of A
    - κ – condition number of A
    - ϵ – accuracy of approximation
    - Up to $\kappa \log(\kappa/\epsilon)$ [2]
- Current implementation using Qiskit



1. State preparation

2. Quantum Phase Estimation

3. Controlled Rotation of Ancilla Qubit

4. Inverse Quantum Phase Estimation

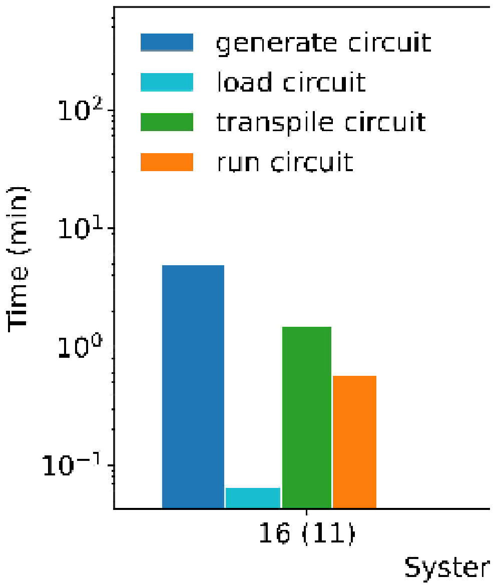
5. Repeat till Ancilla Qubit is 1

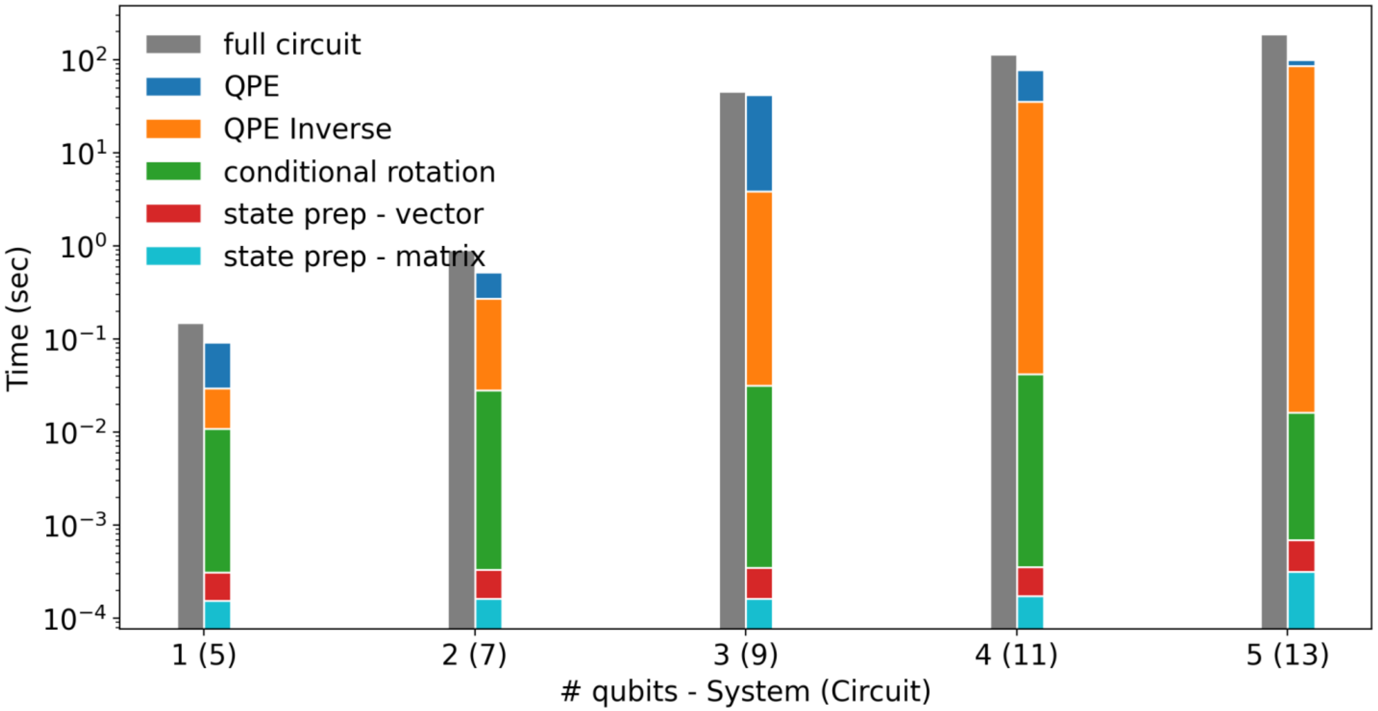
[1] Harrow, Hassidim, and Lloyd, *PRL*, 2009
[2] Morales et al., 2025

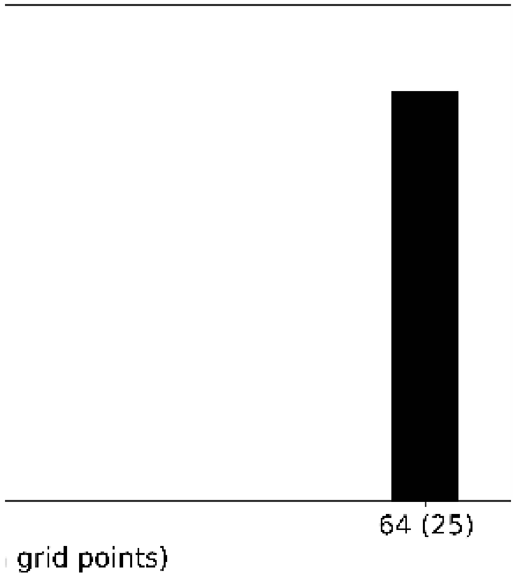
Computational costs are quite high attributing to various components of the problem and algorithm

Testing a sample system of linear equations: tridiagonal Toeplitz matrix

- Need for GPU accelerated simulators: cuQuantum
- Preconditioning matrix or preconditioning-free HHL algorithm
- **Better QPE algorithm**

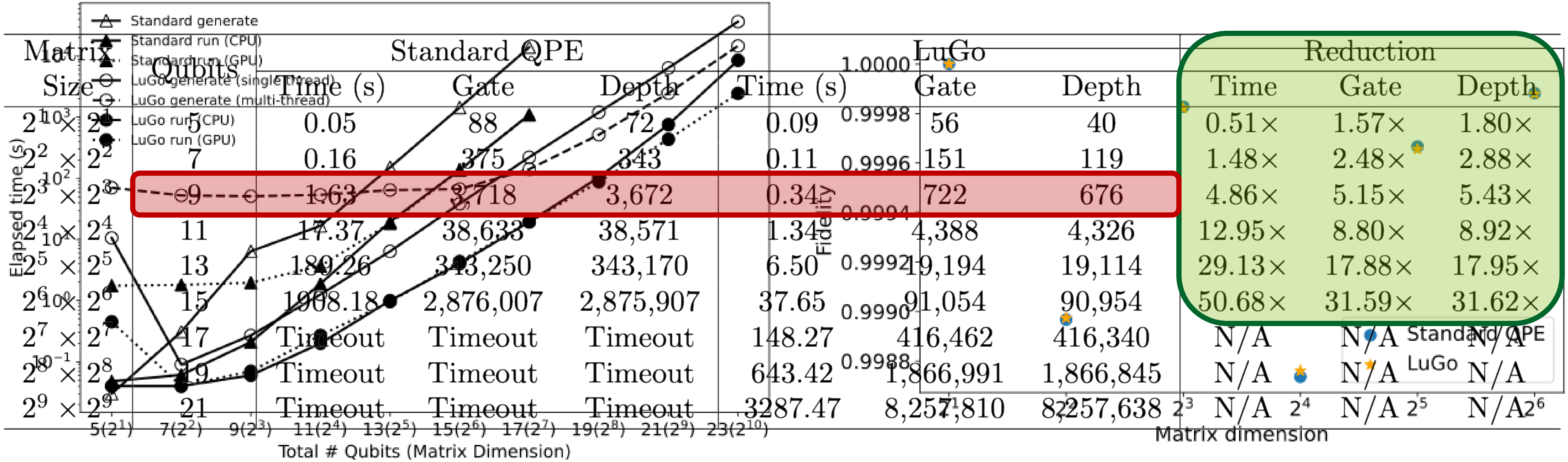






LuGo: An implementation of QPE to eliminate redundant circuit repetitions

- A parallel framework to avoid the exponential growth of controlled-U circuit
- Complexity: Standard - $\mathcal{O}(2^k\mathcal{C}(U))$ LuGo: $\mathcal{O}(k\mathcal{C}(U))$
- LuGo achieves reduction for: (1) time to generate and run circuits, and (2) circuit depth



Matrix Size	Qubits	Standard QPE Time (s)	Standard QPE Gate	Standard QPE Depth	LuGo Time (s)	LuGo Gate	LuGo Depth	Reduction Time	Reduction Gate	Reduction Depth
$2^1 \times 2^1$	5	0.05	88	72	0.09	56	40	0.51×	1.57×	1.80×
$2^2 \times 2^2$	7	0.16	375	343	0.11	151	119	1.48×	2.48×	2.88×
$2^3 \times 2^3$	9	1.63	3,718	3,672	0.34	722	676	4.86×	5.15×	5.43×
$2^4 \times 2^4$	11	17.37	38,633	38,571	1.34	4,388	4,326	12.95×	8.80×	8.92×
$2^5 \times 2^5$	13	189.26	343,250	343,170	6.50	19,194	19,114	29.13×	17.88×	17.95×
$2^6 \times 2^6$	15	1908.18	2,876,007	2,875,907	37.65	91,054	90,954	50.68×	31.59×	31.62×
$2^7 \times 2^7$	17	Timeout	Timeout	Timeout	148.27	416,462	416,340	N/A	N/A	N/A
$2^8 \times 2^8$	19	Timeout	Timeout	Timeout	643.42	1,866,991	1,866,845	N/A	N/A	N/A
$2^9 \times 2^9$	21	Timeout	Timeout	Timeout	3287.47	8,257,810	8,257,638	N/A	N/A	N/A

C. Lu et al., 2025 – *in review*

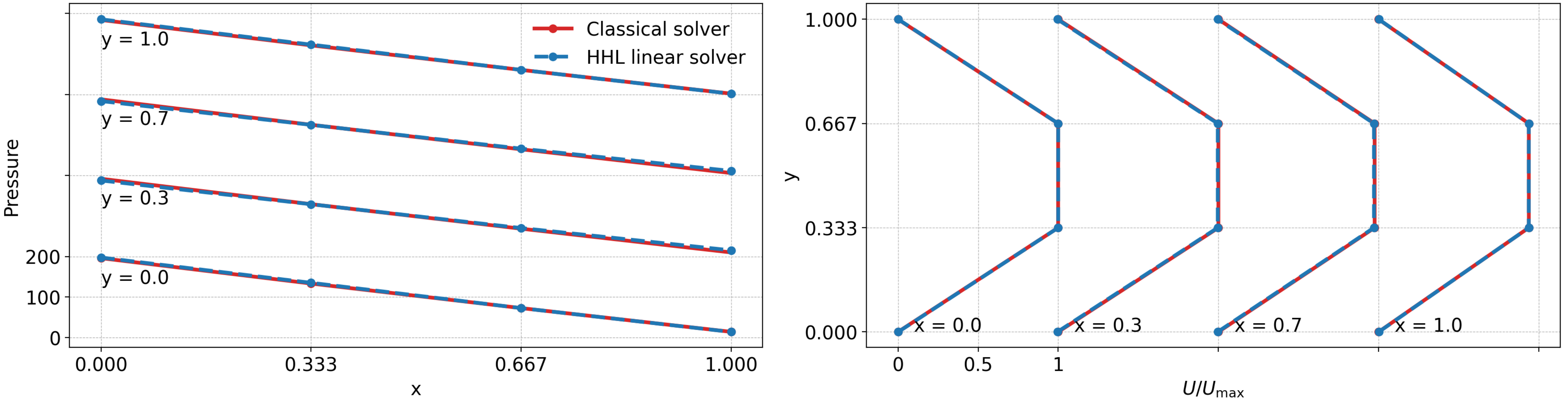
This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231.

Results

Simulator → Emulator → Real Hardware

Validation: Accurate reconstruction of the pressure & velocity profiles achieved using simulators

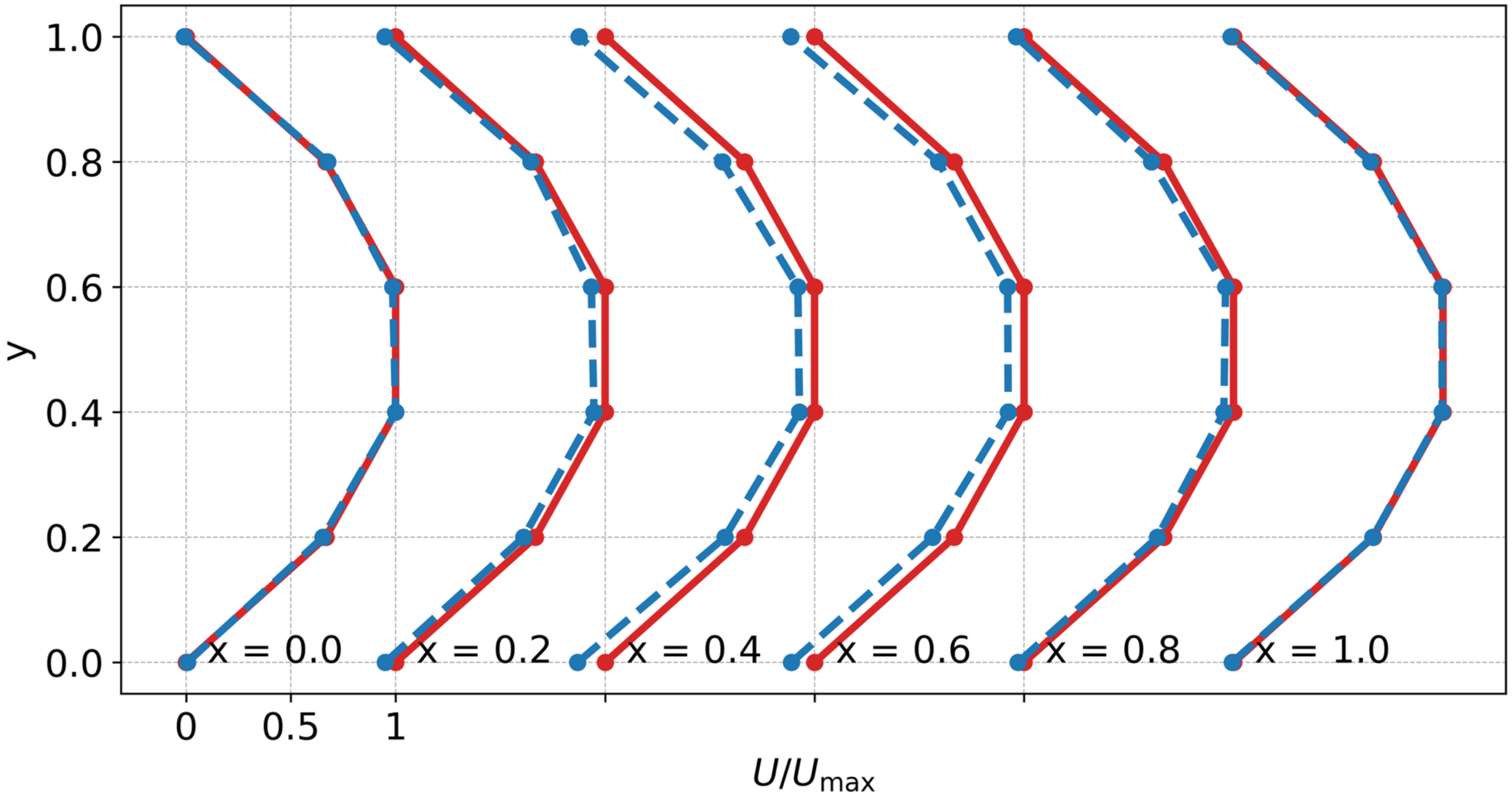
Results using classical simulators



Variable	Pressure	Velocity
Fidelity (%)	99.9	99.9

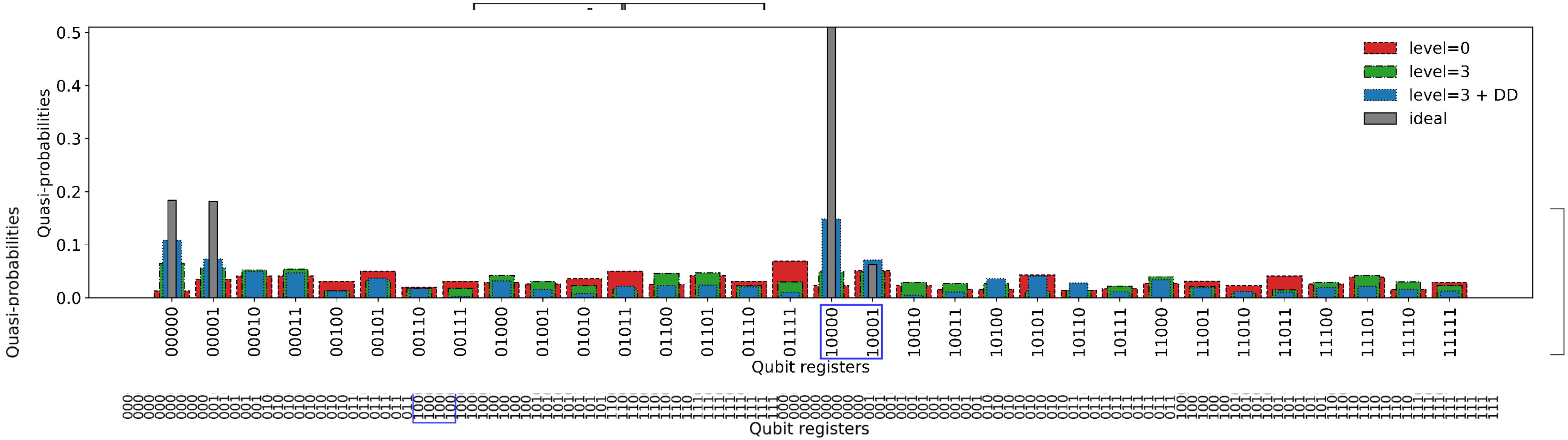
LuGo-based HHL enables scaling to larger flow problems

Results using classical simulators

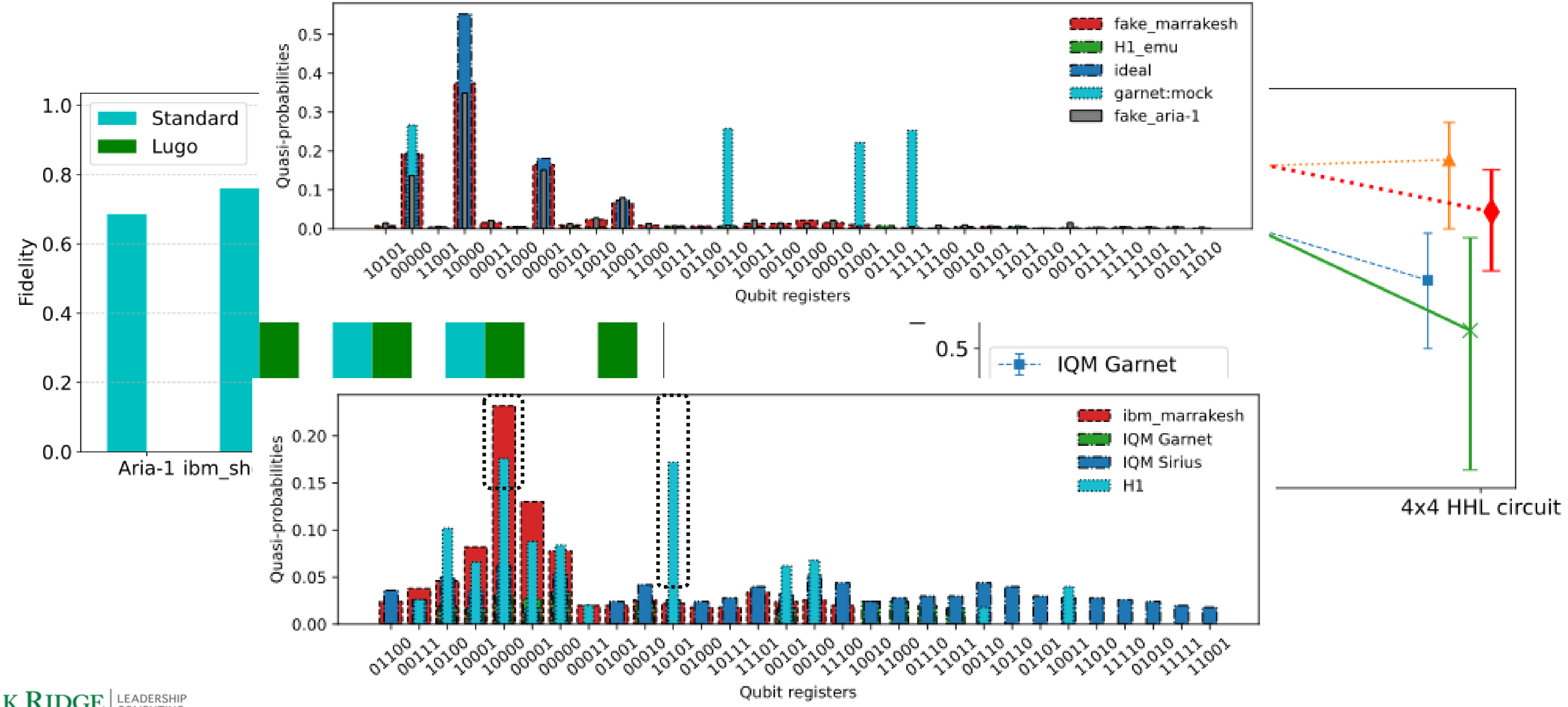


Noise modeling, error mitigation & suppression enable running standard QPE-based HHL on real hardware

- Noise modeling & mitigation using Qiskit primitives: Sampler
- Noise model: *fake* backends
- Error mitigation: qubit readout errors
- Error suppression: Optimizing circuit and Dynamic decoupling

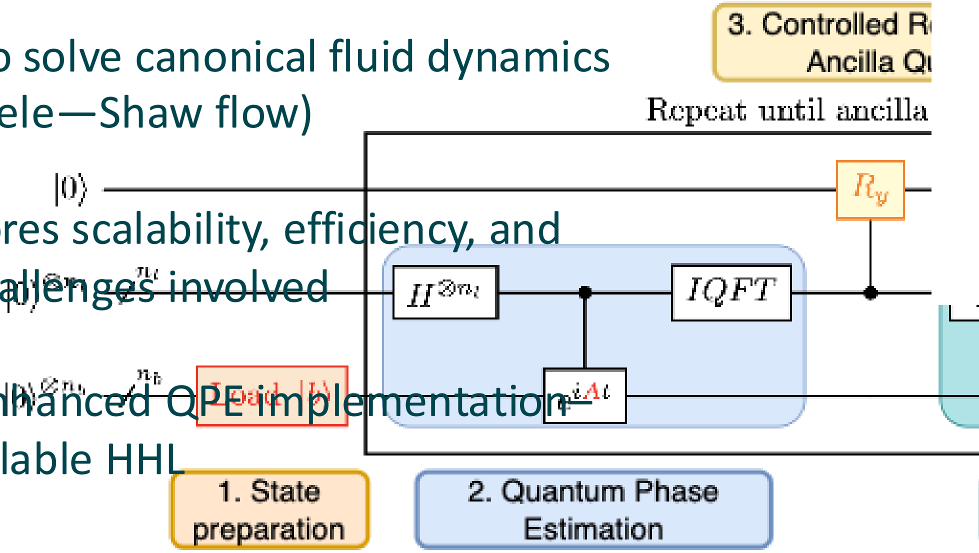


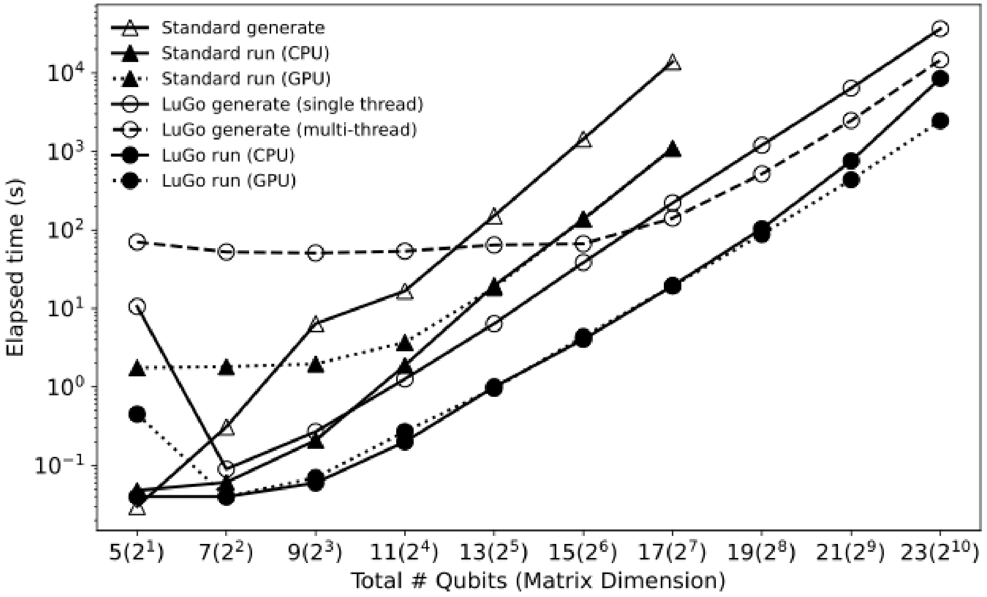
LuGo-based HHL solver better scales on superconducting & trapped-ion quantum hardware

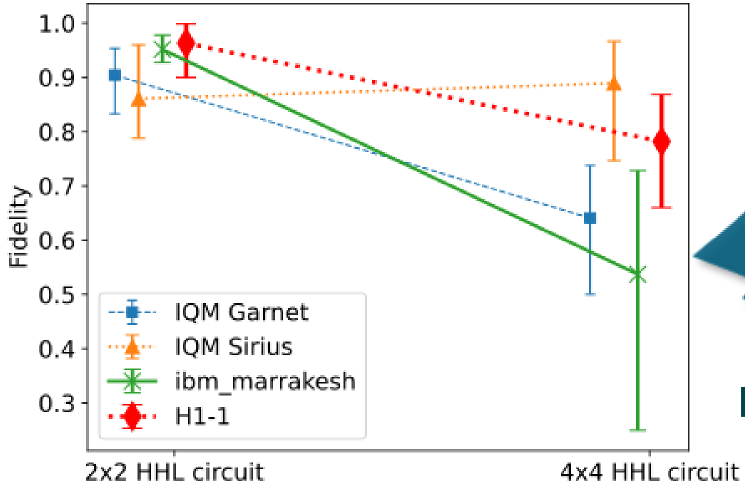


Solving canonical flow problems using the HHL algorithm on superconducting & trapped-ion devices

- A canonical Quantum Linear Systems Algorithm: the Harrow–Hassidim–Lloyd (HHL) algorithm
- HHL used to solve canonical fluid dynamics problem (Hele—Shaw flow)
- Study explores scalability, efficiency, and practical challenges involved
- LuGo–an enhanced QPE implementation–enables scalable HHL







Enabled by QCUP

Gopalakrishnan Meena at al., *PoF*, 2024
Lu et al., 2025a – *in review*
Lu et al., 2025b – *IEEE QCE* 2025

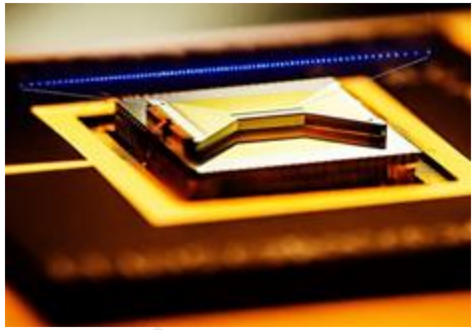
OLCF’s Quantum Computing User Program (QCUP)

- **Premium** access to current stack of **quantum devices** available through OLCF
- Apply any time
- **Free** of charge
- Each project is assigned a **liaison**:
  - ORNL point of contact with quantum science expertise
- Access available for international (non-US) participants

Contact – gopalakrishm@ornl.gov

IBM Quantum

QUANTINUUM



IONQ

IQM

Link

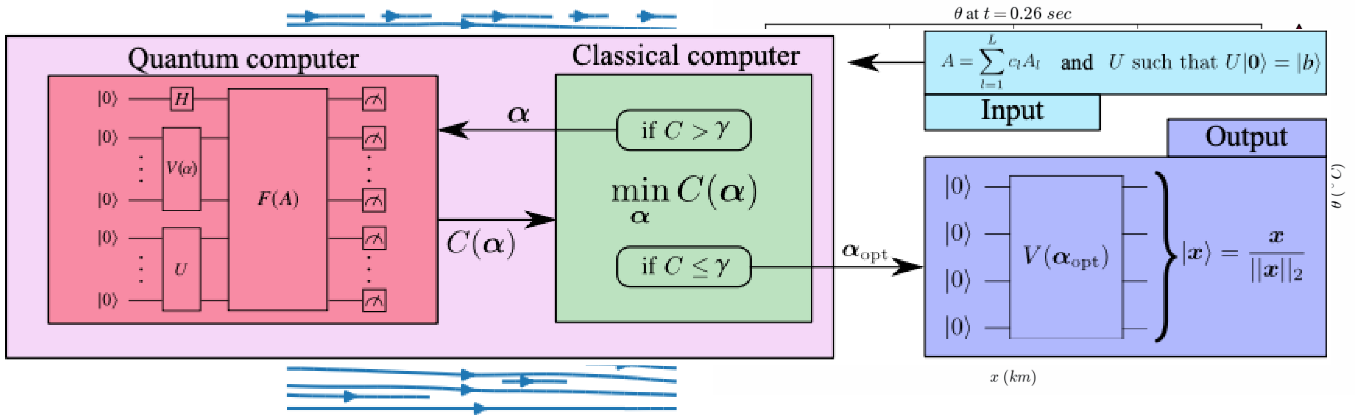
Current & Future directions

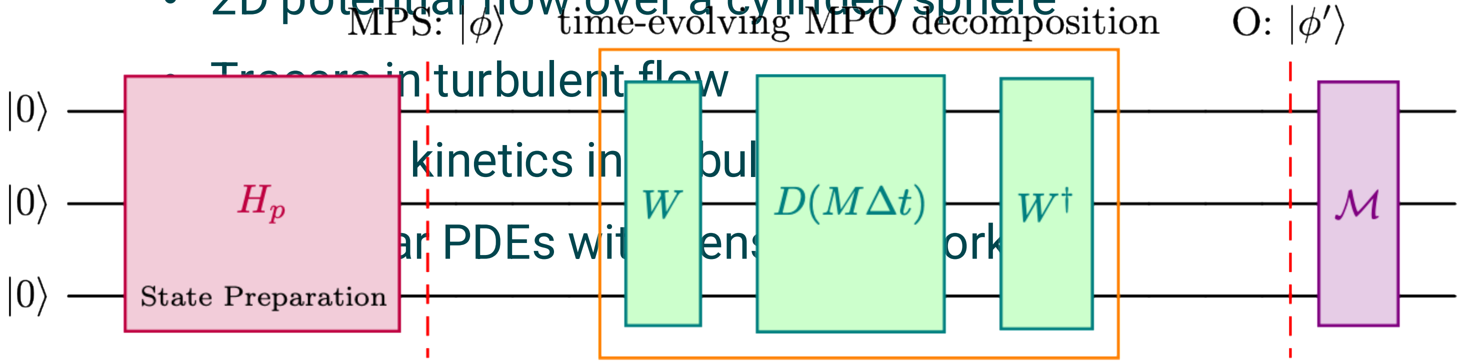
Computational cost

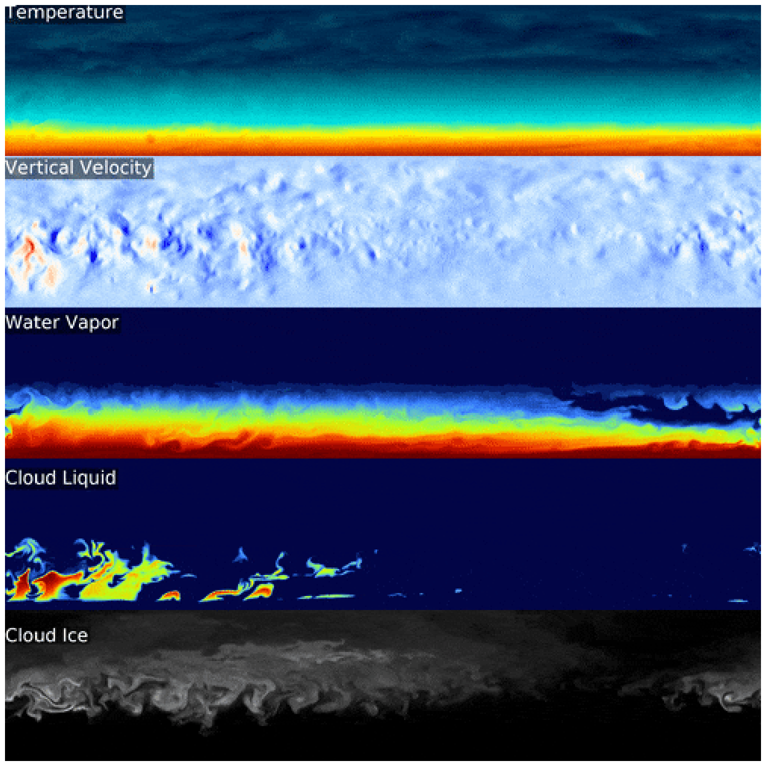
- Need for hybrid quantum-classical algorithms[1,2,3,4,5,6]

Tackle more complex flow problems

- 2D potential flow over a cylinder/sphere
- Tracers in turbulent flow
- kinetics in bul
- ar PDEs wit ens ork







Group website

[1] Bravo-Prieto et al, *Quantum*, 2023
[2] Jaksch et al, *AIAAJ*, 2023
[3] Bharadwaj & Sreenivasan, *PNAS*, 2023
[4] Gopalakrishnan Meena et al., *IEEE QCE*, 2024
[5] Shehata et al., *FGCS* 2025
[6] Gopalakrishnan Meena et al., *IEEE QCE*, 2025
[7] miniWeatherML https://github.com/mrnorman/miniWeatherML

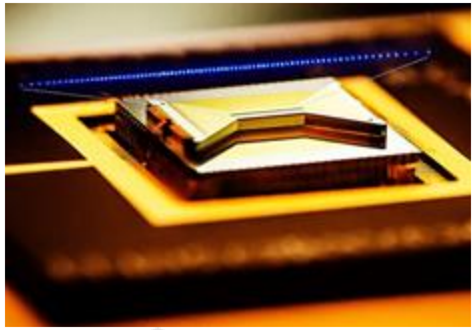
OLCF’s Quantum Computing User Program (QCUP)

- **Premium** access to current stack of **quantum devices** available through OLCF
- Apply any time
- **Free** of charge
- Each project is assigned a **liaison**:
  - ORNL point of contact with quantum science expertise
- Access available for international (non-US) participants

Contact – gopalakrishm@ornl.gov

IBM Quantum

QUANTINUUM



IONQ

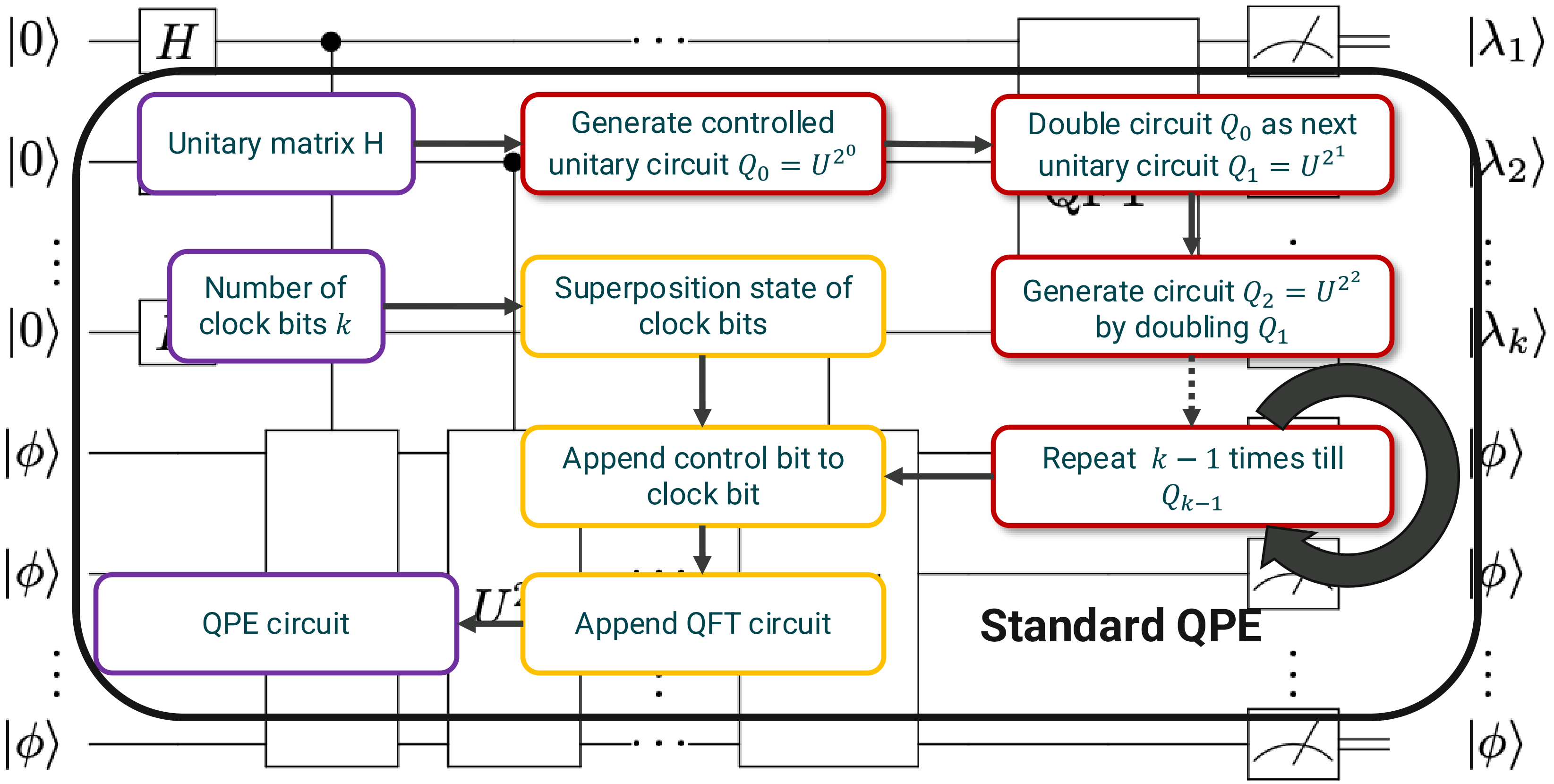
IQM

Link

Thank you

Appendix

An implementation of QPE to eliminate redundant circuit repetitions



Unitary matrix H

Generate controlled unitary circuit $Q_0 = U^{2^0}$

Double circuit Q_0 as next unitary circuit $Q_1 = U^{2^1}$

Number of clock bits k

Superposition state of clock bits

Generate circuit $Q_2 = U^{2^2}$ by doubling Q_1

Append control bit to clock bit

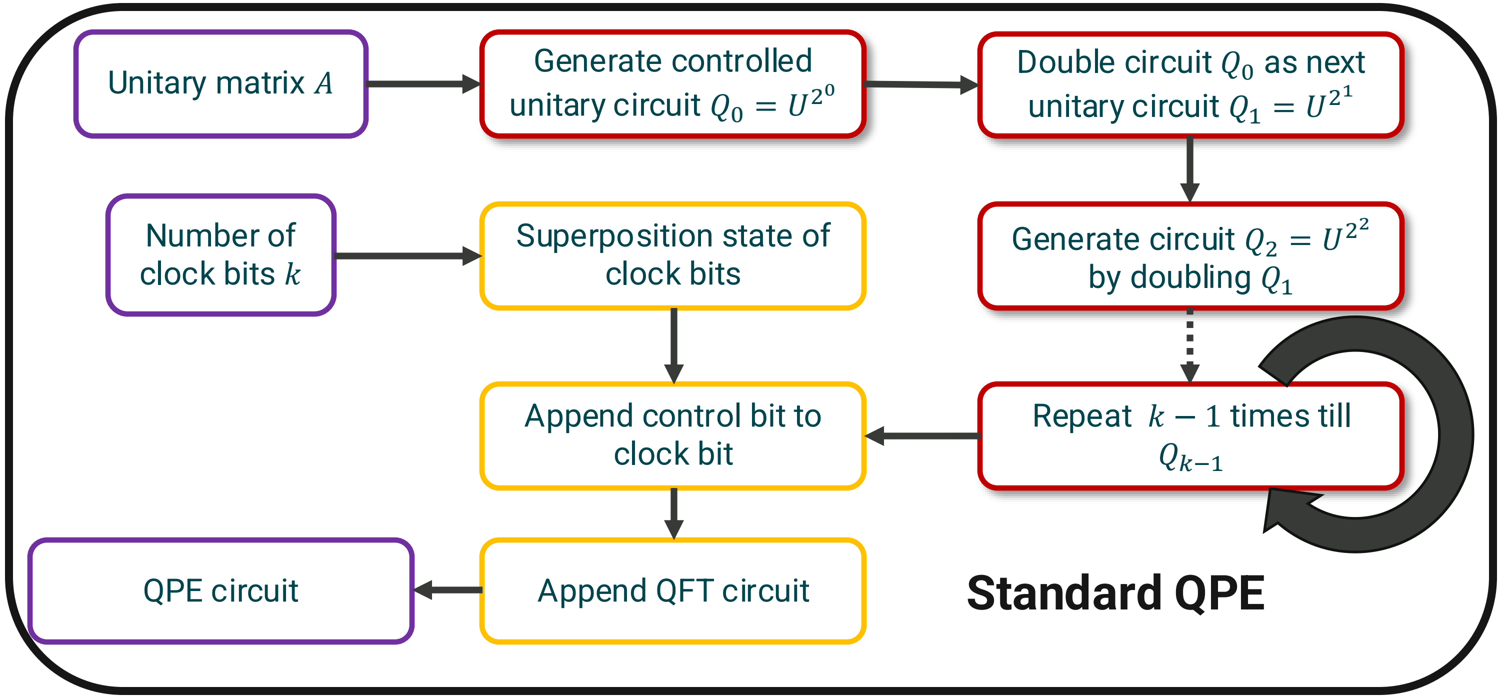
Repeat $k-1$ times till Q_{k-1}

QPE circuit

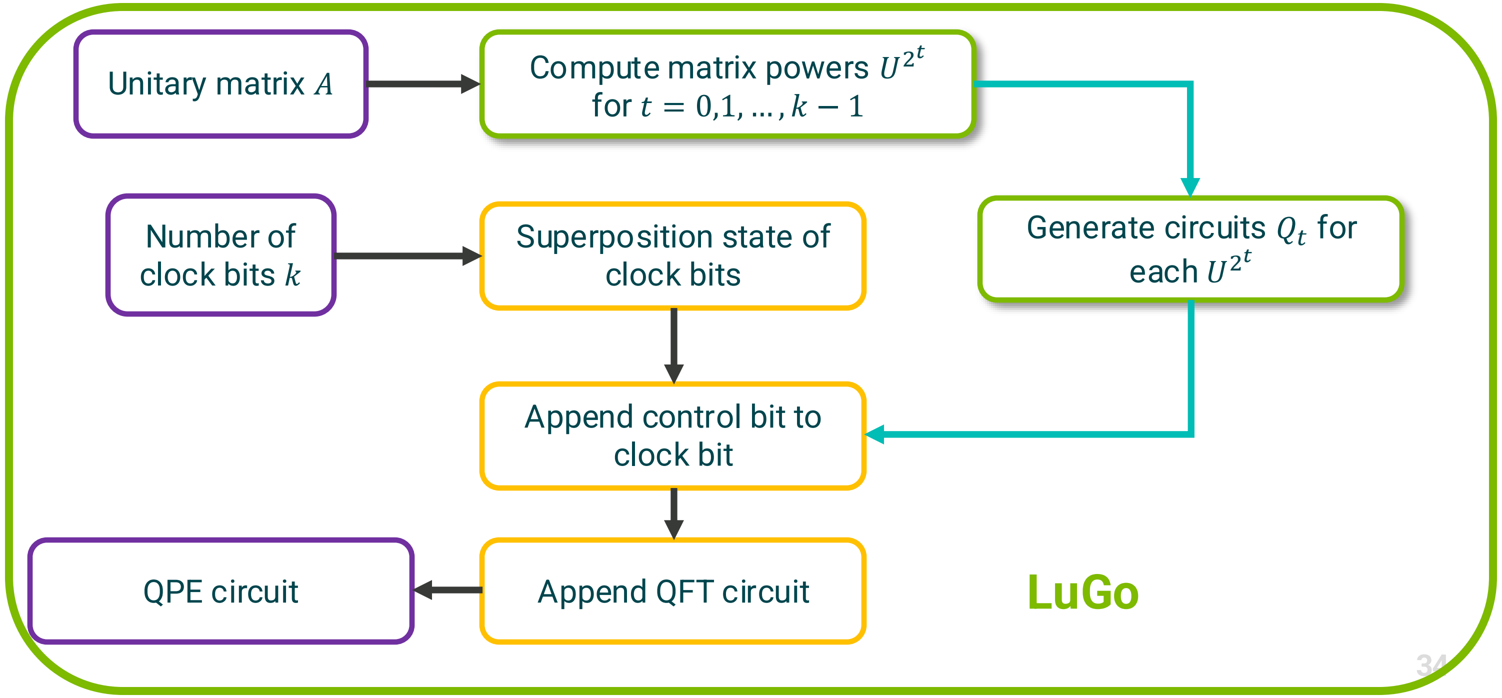
Append QFT circuit

- A parallel framework to avoid the exponential growth of controlled-U circuit (c-U)
- **Each $U_t = e^{iA2^t}$ is computed classically instead of repeating the $U_0 = e^{iA}$ circuit 2^t times**
- Computation of each c-U circuit: embarrassingly parallel - leverages HPC
- Complexity:
  - Standard: $\mathcal{O}(2^k \mathcal{C}(U))$
  - LuGo: $\mathcal{O}(k\mathcal{C}(U))$
- Reduction in circuit depth by minimizing iterations & optimizing design

C. Lu et al., 2025a – *in review*



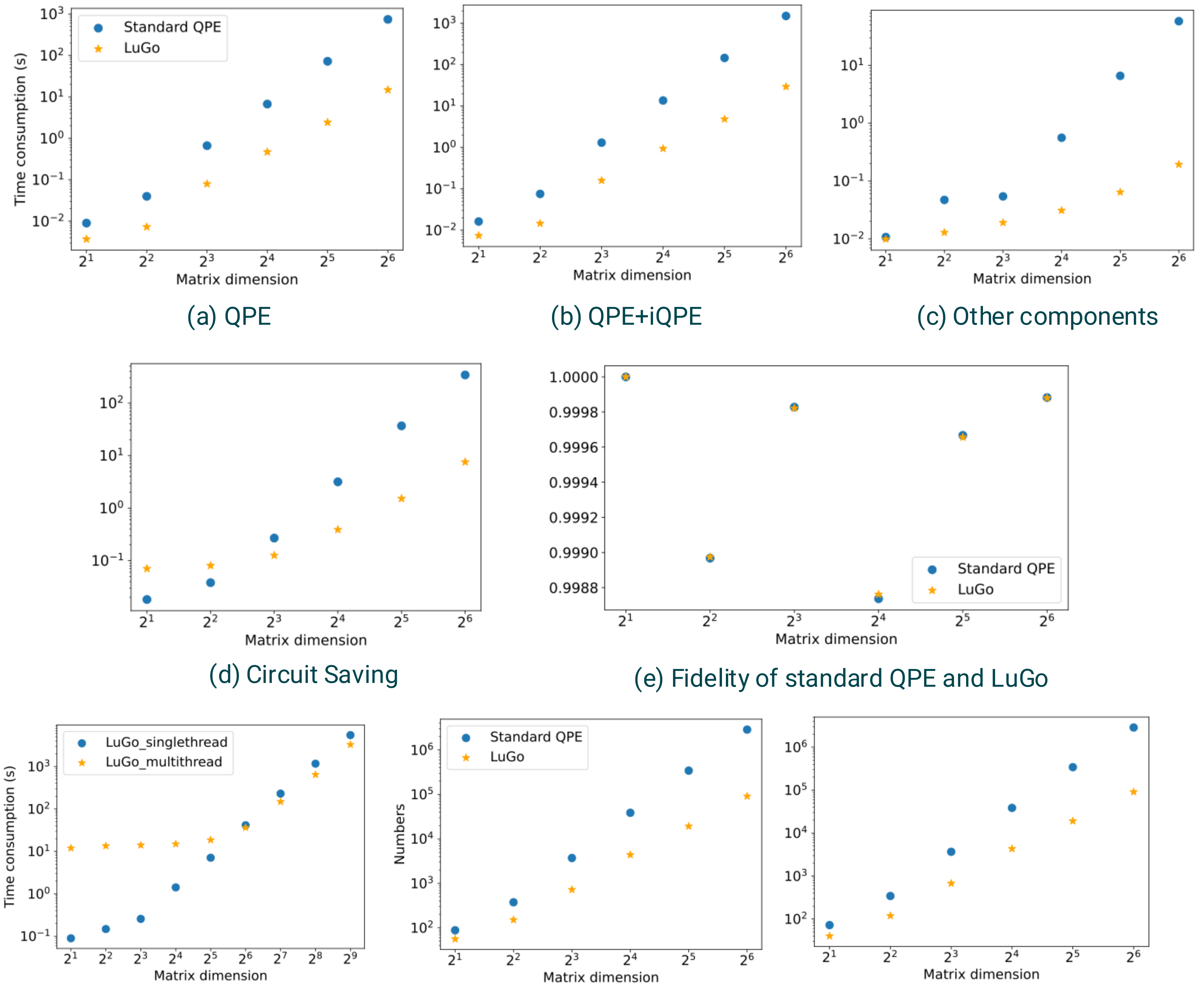
Unitary matrix A
Generate controlled unitary circuit $Q_0 = U^{2^0}$
Double circuit Q_0 as next unitary circuit $Q_1 = U^{2^1}$
Number of clock bits k
Superposition state of clock bits
Generate circuit $Q_2 = U^{2^2}$ by doubling Q_1
Append control bit to clock bit
Repeat $k-1$ times till Q_{k-1}
QPE circuit
Append QFT circuit
Standard QPE



Unitary matrix A
Compute matrix powers U^{2^t} for $t = 0,1,\ldots,k-1$
Number of clock bits k
Superposition state of clock bits
Generate circuits Q_t for each U^{2^t}
Append control bit to clock bit
QPE circuit
Append QFT circuit
LuGo

Results

- We divided total time consumption of HHL to QPE, QPE+iQPE, other components, and circuit saving.
- From the figures, LuGo has better scalability and performance compared to standard QPE generation algorithm.
- LuGo has identical fidelity with standard QPE on ideal Simulator.
- LuGo also obtained circuit compression on circuit count and depth to reduce computing pressure on quantum computers.



(a) QPE

(b) QPE+iQPE

(c) Other components

(d) Circuit Saving

(e) Fidelity of standard QPE and LuGo

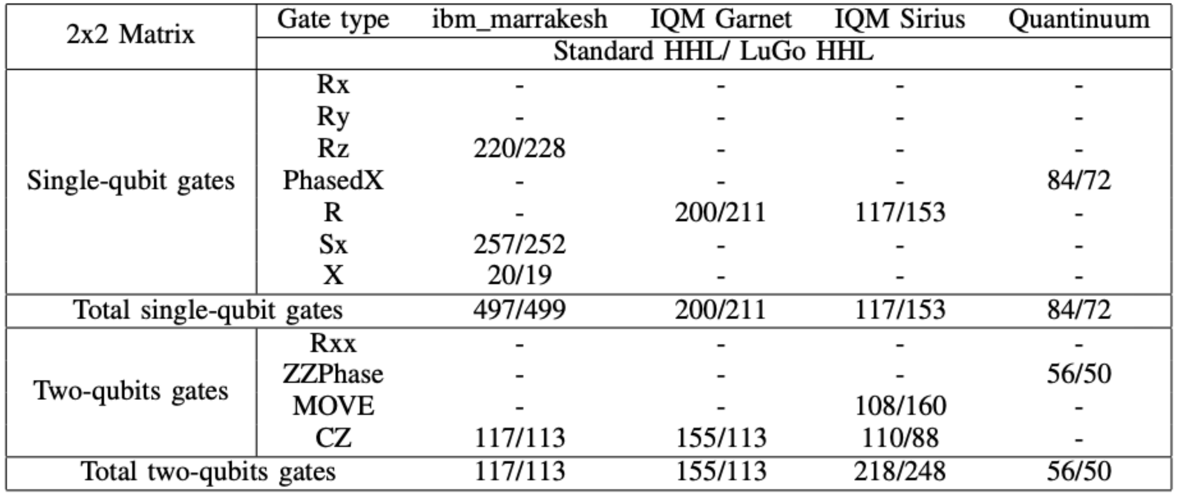
(F) LuGo Multithreading

(G) Gate count

(H) Circuit depth

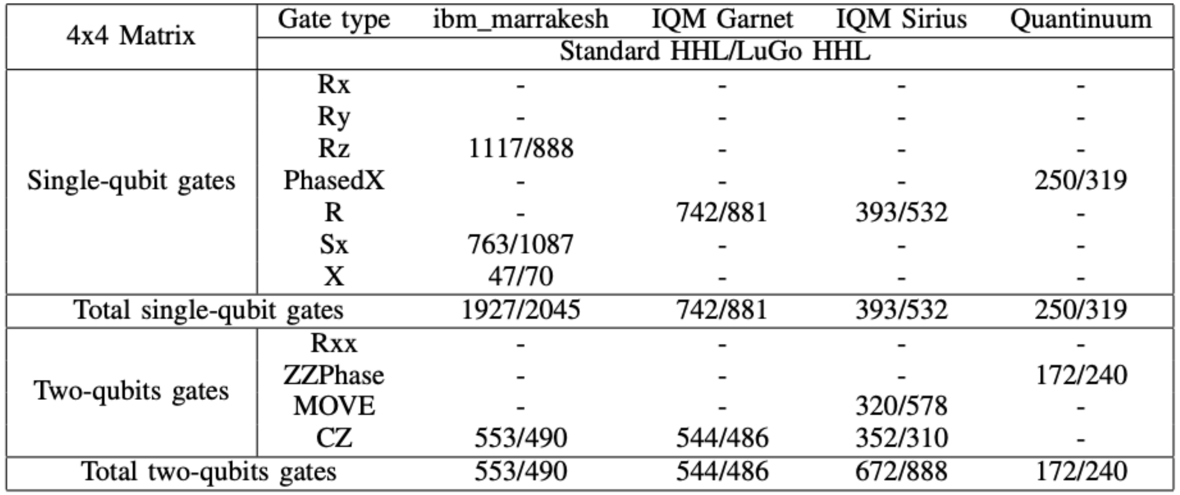
Circuit characteristics of LuGo enabled HHL circuits for superconducting and trapped-ion quantum hardware

2x2 matrix



2x2 Matrix	Gate type	ibm_marrakesh	IQM Garnet	IQM Sirius	Quantinuum
	Standard HHL/ LuGo HHL				
Single-qubit gates	Rx	-	-	-	-
	Ry	-	-	-	-
	Rz	220/228	-	-	-
	PhasedX	-	-	-	84/72
	R	-	200/211	117/153	-
	Sx	257/252	-	-	-
	X	20/19	-	-	-
Total single-qubit gates		497/499	200/211	117/153	84/72
Two-qubits gates	Rxx	-	-	-	-
	ZZPhase	-	-	-	56/50
	MOVE	-	-	108/160	-
	CZ	117/113	155/113	110/88	-
Total two-qubits gates		117/113	155/113	218/248	56/50

4x4 matrix



4x4 Matrix	Gate type	ibm_marrakesh	IQM Garnet	IQM Sirius	Quantinuum
	Standard HHL/LuGo HHL				
Single-qubit gates	Rx	-	-	-	-
	Ry	-	-	-	-
	Rz	1117/888	-	-	-
	PhasedX	-	-	-	250/319
	R	-	742/881	393/532	-
	Sx	763/1087	-	-	-
	X	47/70	-	-	-
Total single-qubit gates		1927/2045	742/881	393/532	250/319
Two-qubits gates	Rxx	-	-	-	-
	ZZPhase	-	-	-	172/240
	MOVE	-	-	320/578	-
	CZ	553/490	544/486	352/310	-
Total two-qubits gates		553/490	544/486	672/888	172/240

OAK RIDGE National Laboratory | LEADERSHIP COMPUTING FACILITY