इलेक्ट्रॉनिकी एवं सूचना प्रौद्योगिकी मंत्रालय
MINISTRY OF ELECTRONICS AND INFORMATION TECHNOLOGY

विज्ञान एवं प्रौद्योगिकी विभाग
DEPARTMENT OF SCIENCE & TECHNOLOGY

सी-डैक
CDAC

Technical Co-Sponsors

IEEE COMPUTER SOCIETY
Bangalore Chapter

IEEE
BANGALORE SECTION

SUPER COMPUTING INDIA 2025

DECEMBER 09 – 13, 2025
Manipal Institute of Technology (MIT)
Bengaluru - KA - India

ORGANISED BY
Centre for Development of Advanced Computing (C-DAC)

Powering the Future

HPC • AI • QUANTUM

https://sci25.supercomputingindia.org

Hybrid Quantum Linear Solver: A hands on tutorial with Qiskit

Rajas Dalvi and Viraj Dsouza

BosonQ Psi (BQP)

Bengaluru, KA, India

December 9, 2025

Outline

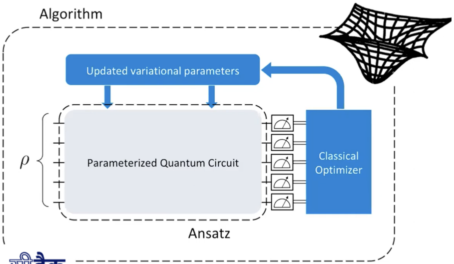
Session Plan & Logistics

- Two sessions
  - Session 1 — Traditional VQLS + hands-on with Qiskit
  - Session 2 — Block-encoding VQLS + demos + discussion
- What to expect today:
  - Motivation & intuition behind VQLS
  - Building VQLS step-by-step with hands-on exercises
  - Block-encoding-based VQLS introduction
- Your turn: try, break, debug, question, explore
- GitHub repo:
  `https://github.com/virajd98/Tutorial_SCI2025-Hybrid_QLS_for_CFD`

Let's get started

Motivation

- Many real-world problems require solving large linear systems $Ax = b$
  - Different domains: CFD, Structural engineering, Machine learning, Electromagnetism, Quantum Chemistry, etc
- Variational quantum linear solvers (VQLS) promise potentially favorable scaling
  - VQLS takes a variational quantum ansatz to prepare a solution state encoding x (up to normalization) by minimizing a suitable cost function.



Linear systems appear in CFD

- CFD problems begin with partial differential equations (PDEs):

  Navier–Stokes, Poisson, Heat equation, Advection–Diffusion, ...

- Spatial discretization (finite difference, finite volume, finite element) converts these PDEs into **large sparse linear systems**.
- Example: discretizing the Poisson equation $-\nabla^2 u = f$ on an $n_x \times n_y$ grid leads to:

$$Au = b, \qquad A \in \mathbb{R}^{N \times N}, \quad N = n_x n_y.$$

- For realistic meshes, A may have dimension 10^6–10^{10}.

Refresher

- A quantum register of n qubits encodes a **normalized vector** in $\mathbb{C}^{2^n}$.
- An operation U on a quantum state is **unitary** $(U^\dagger U = \mathbb{I})$; measurement in computational basis produces bit-strings with probabilities given by squared amplitudes.
- We can prepare parametrized states $|x(\theta)\rangle = V(\theta)\,|0^n\rangle$ and vary θ to minimize a cost.

Refresher: 1-qubit example

- Consider a 1-qubit normalized state

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}, \qquad |\alpha|^2 + |\beta|^2 = 1$$

Let $\alpha = \sqrt{0.7}, \qquad \beta = e^{i\pi/4}\sqrt{0.3}$

Measurement probabilities: $P(0) = |\alpha|^2 = 0.7, \qquad P(1) = |\beta|^2 = 0.3$

- Apply a Parametrized gate (rotation around Y) to $|0\rangle$:

$$R_y(\theta) = \begin{pmatrix}\cos(\theta/2) & -\sin(\theta/2)\\ \sin(\theta/2) & \cos(\theta/2)\end{pmatrix},$$

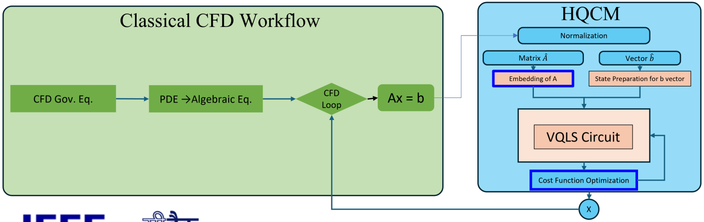
Parametrized state (ansatz): $|\psi(\theta)\rangle = R_y(\theta)|0\rangle = \cos(\frac{\theta}{2})|0\rangle + \sin(\frac{\theta}{2})|1\rangle$

Part I: General framework for VQLS

Workflow

- Unknown solution encoded as $|x(\theta)\rangle = V(\theta)\,|0^n\rangle$- a parametrized quantum state (also called ansatz).
  - $|x(\theta)\rangle$ is a $2^n \times 1$ vector. A is a $N \times N$ matrix where $N = 2^n$.
- Prepare quantum state $|b\rangle$ (assume an efficient short-depth U such that $U\,|0^n\rangle = |b\rangle$).
- Goal: find parameters θ such that $A\,|x(\theta)\rangle \propto |b\rangle$.

Key idea: Convert the classical problem $Ax = b$ into a *Hamiltonian minimization problem*- a quantum problem.



Global Hamiltonian

- If $A|x\rangle \propto |b\rangle$, then the vector $A|x\rangle$ lies entirely in the subspace spanned by $|b\rangle$.
- To measure "how far" $A|x\rangle$ is from $|b\rangle$, project onto orthogonal subspace using $(I - |b\rangle\langle b|)$.
- Define the *Hamiltonian*

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

- Then the unnormalized cost is $\widehat{C}_G(\theta) = \langle x| H_G |x\rangle$.

Computing an expectation value: simple example

Consider a 1-qubit normalized state

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \qquad |\alpha|^2 + |\beta|^2 = 1$$

and a Hermitian matrix (observable)

$$H = \begin{pmatrix} a & c \\ c^* & b \end{pmatrix}, \qquad a, b \in \mathbb{R},\ c \in \mathbb{C}$$

Expectation value:

$$\langle\psi|H|\psi\rangle = \begin{pmatrix} \alpha^* & \beta^* \end{pmatrix} \begin{pmatrix} a & c \\ c^* & b \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

Expanding:

$$\langle H\rangle = a|\alpha|^2 + b|\beta|^2 + c\,\alpha^*\beta + c^*\,\alpha\beta^* \in \mathbb{R}$$

Normalized Global Cost

Based on Bravo et al. [1]

$$\widehat{C}_G(\theta) = \langle x| A^\dagger (I - |b\rangle\langle b|) A |x\rangle \,,$$

$$C_G(\theta) = \frac{\widehat{C}_G(\theta)}{\langle\psi|\psi\rangle} \quad \text{where} \quad |\psi\rangle := A|x\rangle \,.$$

$C_G(\theta)$ is the normalized cost.

$$C_G(\theta) = 1 - \frac{|\langle b| A |x(\theta)\rangle|^2}{\langle x| A^\dagger A |x\rangle}.$$

Interpretation: $C_G(\theta) = 0 \quad \text{iff} \quad A|x\rangle \propto |b\rangle$.

Part II: Traditional VQLS via LCU

Traditional approach: Linear Combination of Unitaries (LCU)

- Assume A is provided as $A = \sum_{\ell=1}^{L} c_\ell A_\ell$ where each A_ℓ is unitary and $c_\ell \in \mathbb{C}$.
- Any matrix can have the above decomposition if A_ℓ represent tensor product of Pauli matrices.
- The four Pauli matrices are:

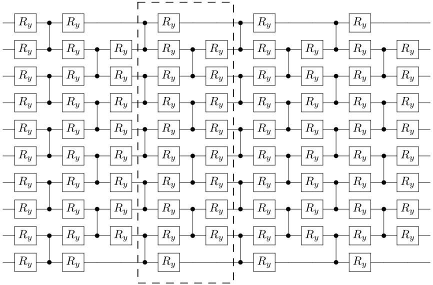
$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Example:

$$\begin{pmatrix} 2 & 0 & 0 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 1 & 2 & 0 \\ 1 & 0 & 0 & 2 \end{pmatrix} = 2(I \otimes I) + X \otimes X.$$

Hardware-Efficient Ansatz

- Designed to match the native gate set and connectivity of quantum hardware.
- Alternating layers of single-qubit rotations and entangling gates.
- Parameter count grows linearly with number of qubits and layers.



The dashed box represents a single layer.

1. How many layers are there in the above representative figure?

2. What are the number of angle parameters for an n-qubit HEA with L layers?

Hands-on: LCU and HEA ansatz (15 min)

- Introduction to the repository
- **Exercise 1:** Go through the code for LCU decomposition and try the code for different sizes
- **Exercise 2:** Visualize a 5 qubit HEA with 3 layers on qiskit.

Expand cost in terms of LCU coefficients

Substitute $A = \sum_\ell c_\ell A_\ell$ into numerator and denominator of $C_G(\theta)$:

$$\langle x| A^\dagger A |x\rangle = \sum_{\ell,\ell'} c_\ell c_{\ell'}^* \underbrace{\langle x| A_{\ell'}^\dagger A_\ell |x\rangle}_{\beta_{\ell\ell'}(\theta)},$$

$$|\langle b| A |x\rangle|^2 = \sum_{\ell,\ell'} c_\ell c_{\ell'}^* \underbrace{\langle 0| U^\dagger A_\ell V |0\rangle \langle 0| V^\dagger A_{\ell'}^\dagger U |0\rangle}_{\gamma_{\ell\ell'}(\theta)}.$$

Definitions:

$$\beta_{\ell\ell'}(\theta) := \langle 0| V^\dagger A_{\ell'}^\dagger A_\ell V |0\rangle, \quad \gamma_{\ell\ell'}(\theta) := \langle 0| U^\dagger A_\ell V |0\rangle \langle 0| V^\dagger A_{\ell'}^\dagger U |0\rangle.$$

These β and γ are computed via appropriate quantum circuits.

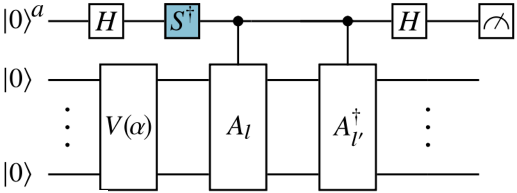
Estimating the Cost Coefficients with Low-Depth Circuits

Estimating $\beta_{\ell\ell'}$:

- For $\ell = \ell'$, $\beta_{\ell\ell} = 1$.
- For $\ell \neq \ell'$, use a Hadamard test as shown below: (insert phase gate for the imaginary part)

$$\mathrm{Re}(\beta_{\ell\ell'}) = 2(P(0) - P(1)),$$

$$\beta_{\ell\ell'} = \mathrm{Re}(\beta_{\ell\ell'}) + i\mathrm{Im}(\beta_{\ell\ell'})$$



Estimating the Cost Coefficients with Low-Depth Circuits

Estimating $\gamma_{\ell\ell'}$:

- For $\ell = \ell'$, $\gamma_{\ell\ell} = |\langle 0| U^\dagger A_\ell V |0\rangle|^2$, is directly measurable as the probability of obtaining $|0\rangle$ on the state $U^\dagger A_\ell V |0\rangle$.
- For $\ell \neq \ell'$, use the same Hadamard-test circuit as for $\beta_{\ell\ell'}$.

Hands-on: Computing $\beta_{\ell\ell'}$ and $\gamma_{\ell\ell'}$(30 min)

Exercise 3:

For the HEA ansatz with 1 layer and LCU terms $A_0 = I \otimes I$ and $A_1 = X \otimes X$, compute β_{01}, γ_{01} and γ_{11}. Assume $U = H \otimes H$ and all the angles in the ansatz to be $\frac{\pi}{4}$.

Putting the pieces together: cost evaluation

- Compute all needed $\beta_{\ell\ell'}(\theta)$ and $\gamma_{\ell\ell'}(\theta)$ via Hadamard tests.
- Total quantum circuits needed: $\mathcal{O}(L^2)$
- Assemble numerator $\sum_{\ell,\ell'} c_\ell c_{\ell'}^* \gamma_{\ell\ell'}$ and denominator $\sum_{\ell,\ell'} c_\ell c_{\ell'}^* \beta_{\ell\ell'}$.
- Evaluate

$$C_G(\theta) = 1 - \frac{\sum_{\ell,\ell'} c_\ell c_{\ell'}^* \gamma_{\ell\ell'}(\theta)}{\sum_{\ell,\ell'} c_\ell c_{\ell'}^* \beta_{\ell\ell'}(\theta)}.$$

- Use a classical optimizer to update θ to minimize $C_G(\theta)$.

Hands-on: Cost function optimization (15 min)

Exercise 4

Using Ising model given in the notebook as the A matrix, perform the cost function optimization for problem sizes 32×32 and 256×256.

Operational meaning: cost $\to$ observable error bound

- Cost function is lower bounded:

$$C_G(\theta) \geq \frac{\epsilon^2}{\kappa^2}$$

  κ is the condition-number and ϵ is the allowed error between the obtained and actual solution.
- Practically: choose a tolerance γ s.t. $C_G(\theta) \leq \gamma$ implies the desired classical error ϵ on observables derived from $|x\rangle$.
- Terminate hybrid loop when $C_G(\theta) \leq \gamma$ and set $\theta^* = \arg\min C_G$.

Recovering the solution vector x from θ^*

- At optimum θ^*, $C_G(\theta^*) \approx 0$ implies $A\,|x(\theta^*)\rangle \propto |b\rangle$, so $|x(\theta^*)\rangle$ encodes the solution up to scale.
- Classical recovery options:
  1. **Quantum state tomography** of $|x(\theta^*)\rangle$ (infeasible for large n).
  2. **Estimate desired observables** directly on the prepared state.
- Appropriate classical post processing is used to compute the vector x from the quantum state $|x(\theta^*)\rangle$.

Practical remark: VQLS is most useful for computing expectation values of some useful observables for the state $|x\rangle$, than the full classical vector.

Hands-on: State vector simulation and Post-processing

Exercise 5

Understand the post processing factor to obtain classical x from quantum $|x(\theta)\rangle$

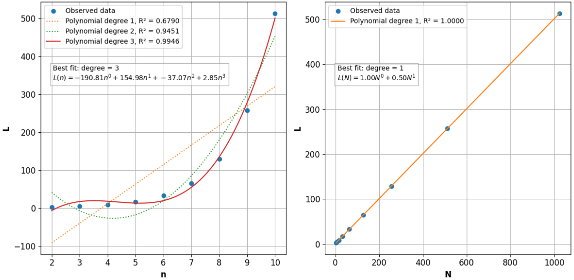
The classical x relates to quantum state $|x(\theta)\rangle$ as given below,

$\boxed{x = \|\vec{x}\| \, |x(\theta)\rangle}$ where,

$$\|\vec{x}\| = \sqrt{\frac{\langle\vec{b}|\,\vec{b}\rangle}{\langle\psi|\,A^{\dagger}A\,|\psi\rangle}}$$

Practical issues for real world problems

- The number of different quantum circuits needed to evaluate the cost for one iteration is $\mathcal{O}(L^2)$. We need $L = \mathcal{O}(poly(log(N))$ which need not be the case in general.



- Time for matrix decomposition
- Barren plateaus

D'Souza et al. 2025[2]

Part III: BQP's VQLS via efficient Block-Encodings

BQP Approach: Block encoding based VQLS

Embedding a matrix $A \in \mathbb{C}^{2^n \times 2^n}$ with $\|A\|_2 \leq 1$ into a larger unitary $U_A \in \mathbb{C}^{2^{m+n} \times 2^{m+n}}$:

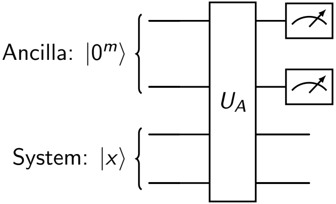
$$U_A = \begin{bmatrix} \frac{A}{\alpha} & * \\ * & * \end{bmatrix}$$

- Input: $|0^m\rangle \otimes |x\rangle$
- Apply U_A, measure ancilla. If result is $|0^m\rangle$, the system is projected to $A|x\rangle$

$$U_A v = \begin{bmatrix} \frac{A}{\alpha} x \\ * \end{bmatrix} = \frac{|0\rangle}{\alpha} A|x\rangle + \sum_{k \neq 0} |k\rangle|*\rangle,$$

- Success probability: $\|A|x\rangle\|^2/\alpha^2$

Visualization



Idea: If we can construct U_A with moderate circuit depth and if the success probability is sufficiently high for matrices of interest, then we can use this circuit inplace of all the $\mathcal{O}(L^2)$ Hadamard test circuits for computing the cost function.

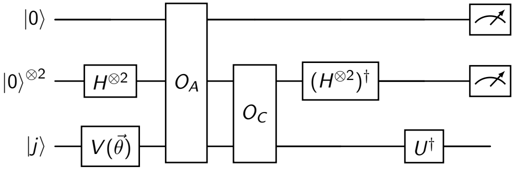
Hands-on: Block encoding a $1D$ Poisson system

Exercise 6

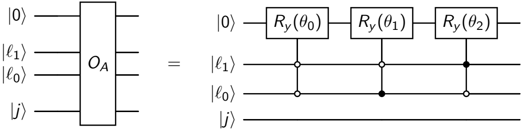
Run the notebook `blocks_demo.ipynb` and understand the results.

Example: VQLS for a $1D$ Poisson system

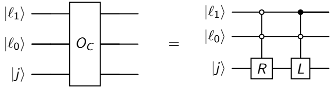
Full circuit:



Unitary O_A



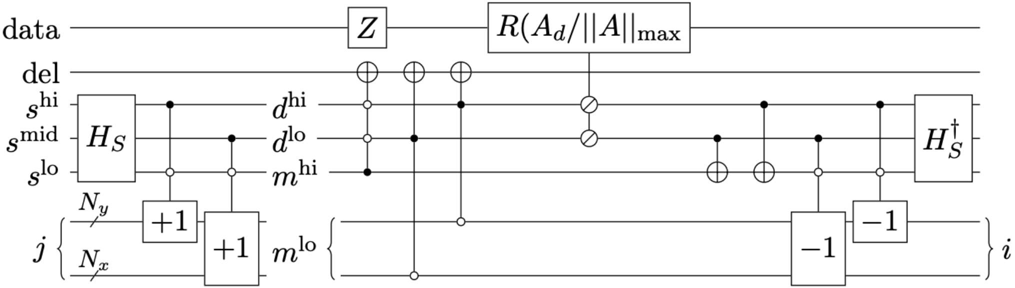
Unitary O_C



(D'Souza et al., 2025 [2], Camps et al., 2023 [3])

Block encoding a $2D$ Laplacian

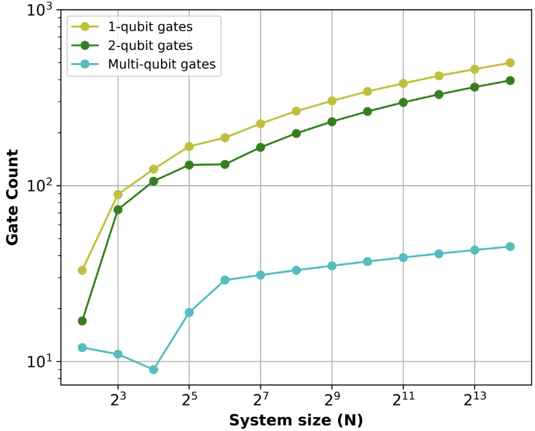
The below circuit can be used to block encode a $2D$ Laplacian matrix (with Dirichlet boundary conditions along each dimensions)



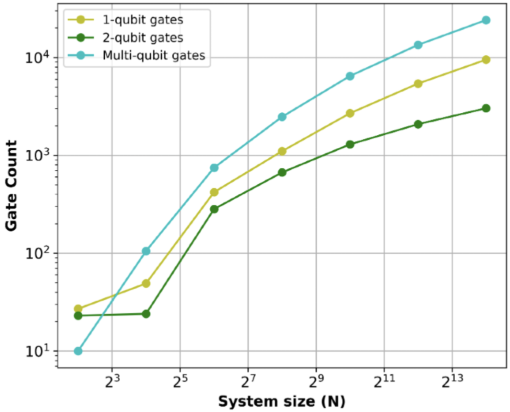
(Sunderhauff et al., 2024 [4])

Resource estimation: VQLS for $1D$ and $2D$ Poisson systems

(D'Souza et al., 2025 [2])



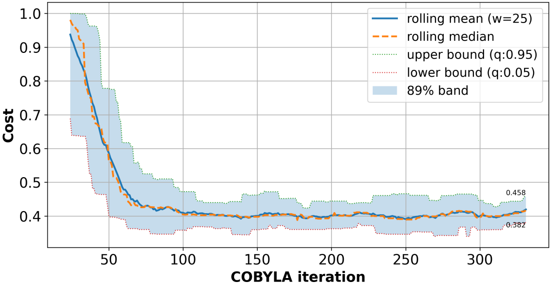
1D Poisson system



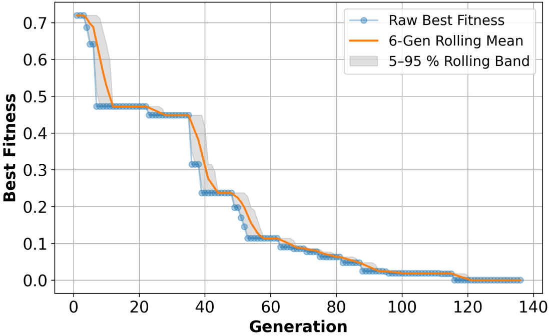
2D Poisson system

VQLS Cost Optimization comparison between COBYLA and QIEO

(D'Souza et al., 2025 [2], Kumar et al., 2024[5])



COBYLA Optimizer



QIEO Optimizer

References

[1] Carlos Bravo-Prieto, Ryan LaRose, Marco Cerezo, Yigit Subasi, Lukasz Cincio, and Patrick J Coles. Variational quantum linear solver. *Quantum*, 7:1188, 2023.

[2] Viraj D'Souza, S. Dhamotharan, Tanishque Kumar, Ramesh Kolluru, Ferdin Don Bosco, Abhishek Chopra, Rut Lineswala, and Onkar Sahni. Towards hybrid quantum-classical computing for large-scale cfd: Tackling scalability challenges of the variational quantum linear solver. In *AIAA AVIATION Forum and ASCEND*, July 2025. AIAA 2025-3042.

[3] Daan Camps, Lin Lin, Roel Van Beeumen, and Chao Yang. Explicit quantum circuits for block encodings of certain sparse matrices, 2023.

[4] Christoph Sünderhauf, Earl Campbell, and Joan Camps. Block-encoding structured matrices for data input in quantum computing. *Quantum*, 8:1226, 2024.

[5] Kandula Eswara Sai Kumar, Rajas Dalvi, Aman Mittal, Aakif Akhtar, Ferdin Don Bosco, Rut Lineswala, Abhishek Chopra, et al. Benchmarking of gpu-optimized quantum-inspired evolutionary optimization algorithm using functional analysis. *arXiv preprint arXiv:2412.08992*, 2024.

Thank you, Questions?

bQp
Do more with less

The World's First Quantum-Powered Simulation Platform