Use of Quantum Computing for CAE Simulation

Prith Banerjee, Ph.D.
Chief Technology Officer, Ansys

TQCI Conference

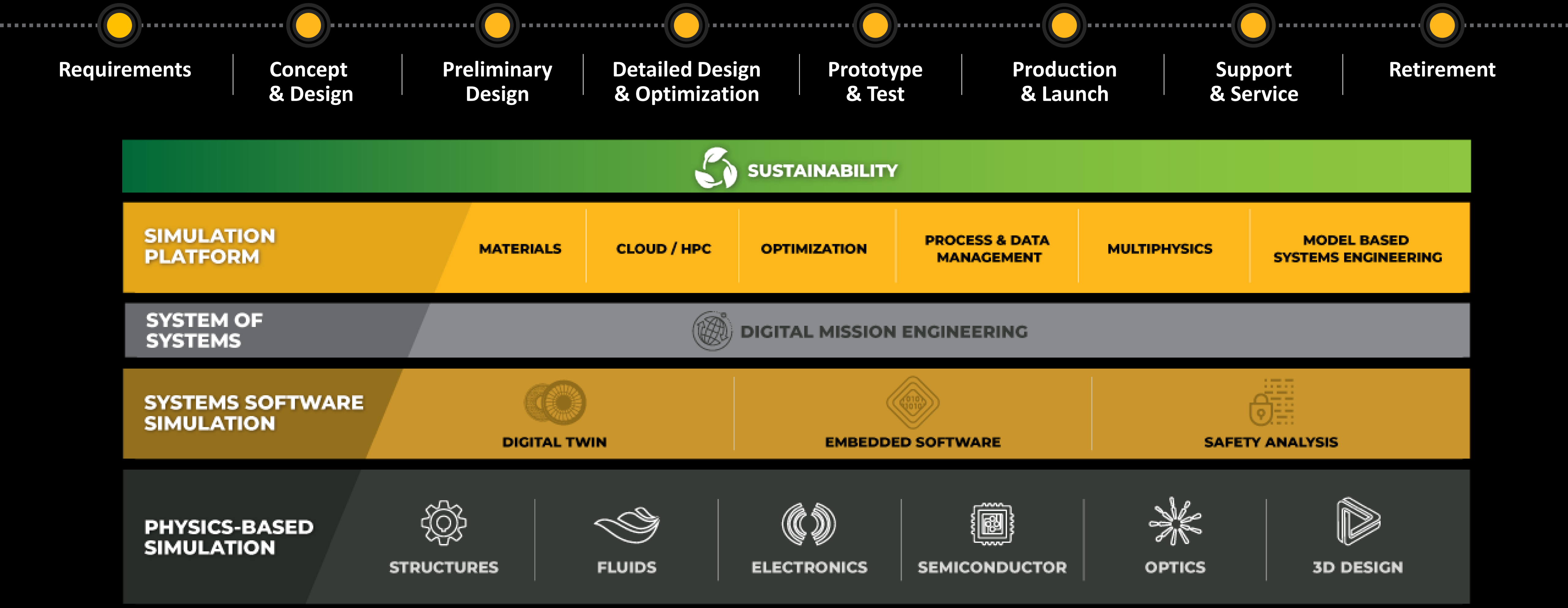
June 2025

Agenda

- Introduction to Ansys
- Use of Quantum Computers to Accelerate CAE Simulation
  - Graph Partitioning
  - Quantum Lattice Boltzmann method for Fluids
  - Hamiltonian Simulation

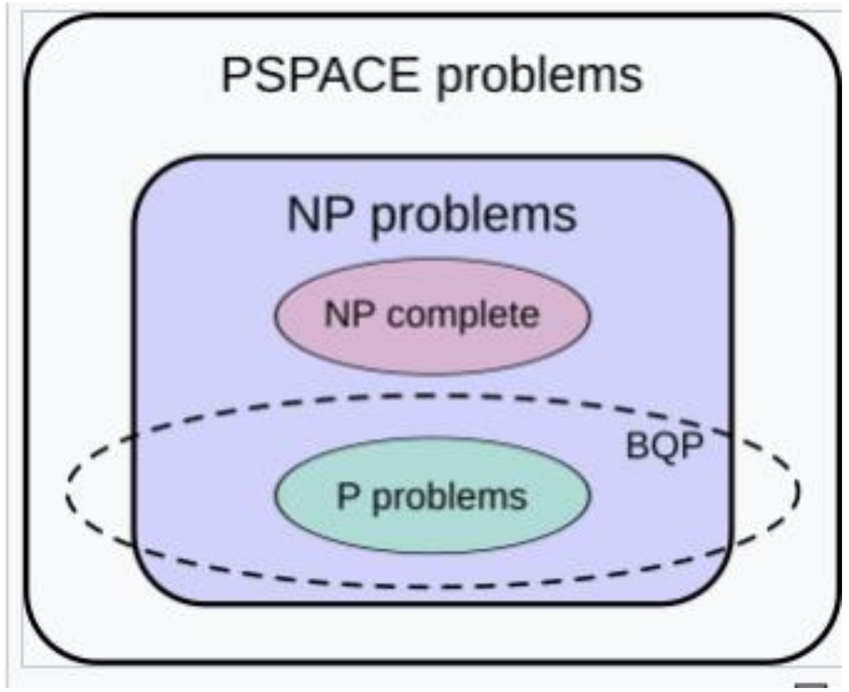
Optimizing Product Development From Component to Maintenance

Unique design of the Ansys product portfolio, platform, and ecosystem for your development processes



Quantum Computing Applications

- Quantum computing provides exponential computing power and are most suitable for problems called BQP (Bounded error Quantum Polynomial time)
- What makes a quantum algorithm faster than a classical one?
  - **Quantum parallelism**: by using superposition the computer is executing on all possible inputs at once
- Applications where Quantum Computing does well
  - Factoring, Searching, Cryptography, Security, Optimization, Scheduling, Graph partitioning, Cell placement, Wire routing, Molecular drug discovery
- It is not obvious how quantum computing can be used to speed up engineering simulation
  - We deal with solution of linear systems of equations, A.x = b (requiring N^3 computations)

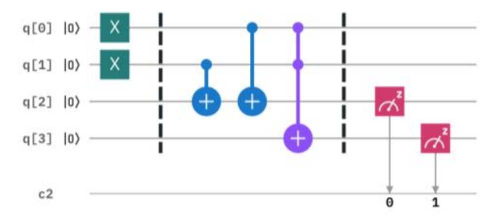


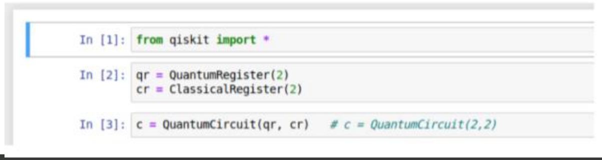
Programming Quantum Computers

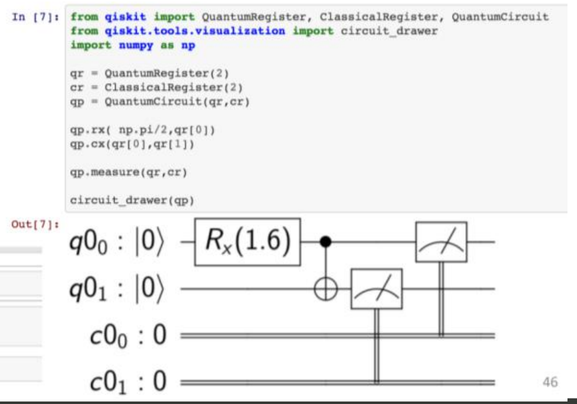
- QISKit from IBM - Python (supports IBM, AQT, IQM, IONQ, Quantinuum, Rigetti)
- CUDA-Q from NVIDIA runs on GPUs (supports IONQ, IQM,Pasqal, Rigetti, Quantinuum, QuEra)
- Q# from Microsoft Azure (supports MSFT, Rigetti, IONQ, Quantinuum, Pasqal, QCI)
- Amazon Braket from AWS (supports IONQ, IQM, Rigetti, QuEra)

Ansys CTO office is developing quantum algorithms in QISKIT and CUDA-Q

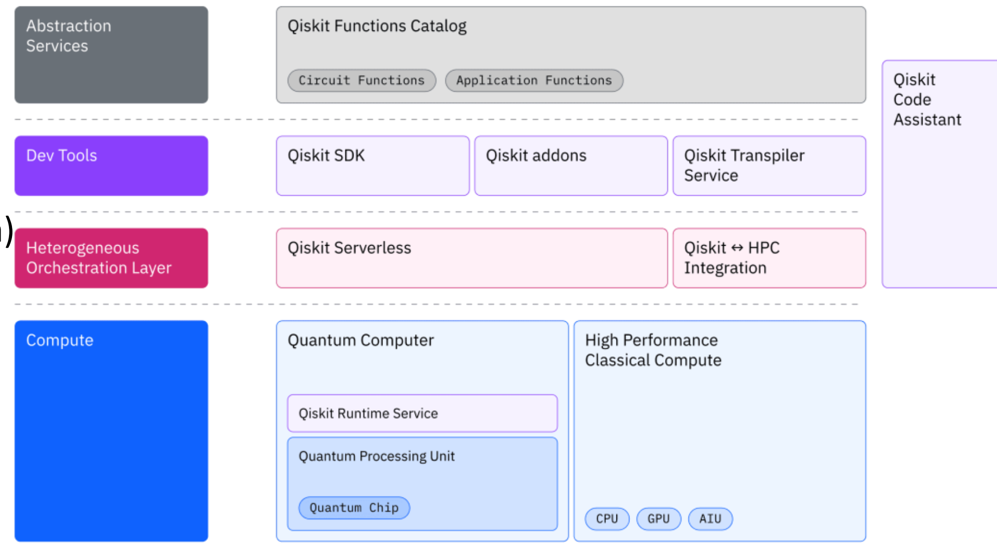
- **<u>QISKit (Quantum Information Science Kit)</u>**
  - QISKit is an open-source framework for quantum computing. It provides tools for creating and manipulating quantum programs and running them on prototype quantum devices over cloud-based access




```
In [1]: from qiskit import *

In [2]: qr = QuantumRegister(2)
        cr = ClassicalRegister(2)

In [3]: c = QuantumCircuit(qr, cr)    # c = QuantumCircuit(2,2)
```

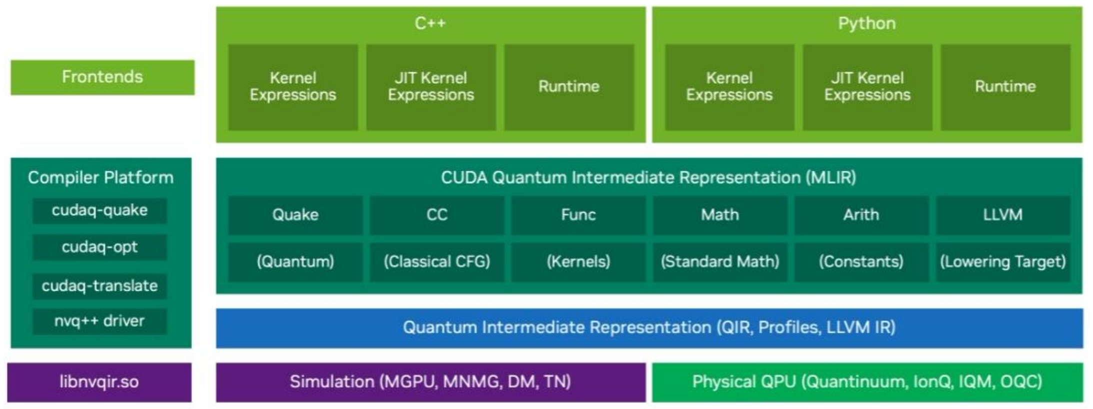

```
In [7]: from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
        from qiskit.tools.visualization import circuit_drawer
        import numpy as np

        qr = QuantumRegister(2)
        cr = ClassicalRegister(2)
        qp = QuantumCircuit(qr,cr)

        qp.rx( np.pi/2,qr[0])
        qp.cx(qr[0],qr[1])

        qp.measure(qr,cr)

        circuit_drawer(qp)

Out[7]:
```

IBM QISKIT Tech Stack

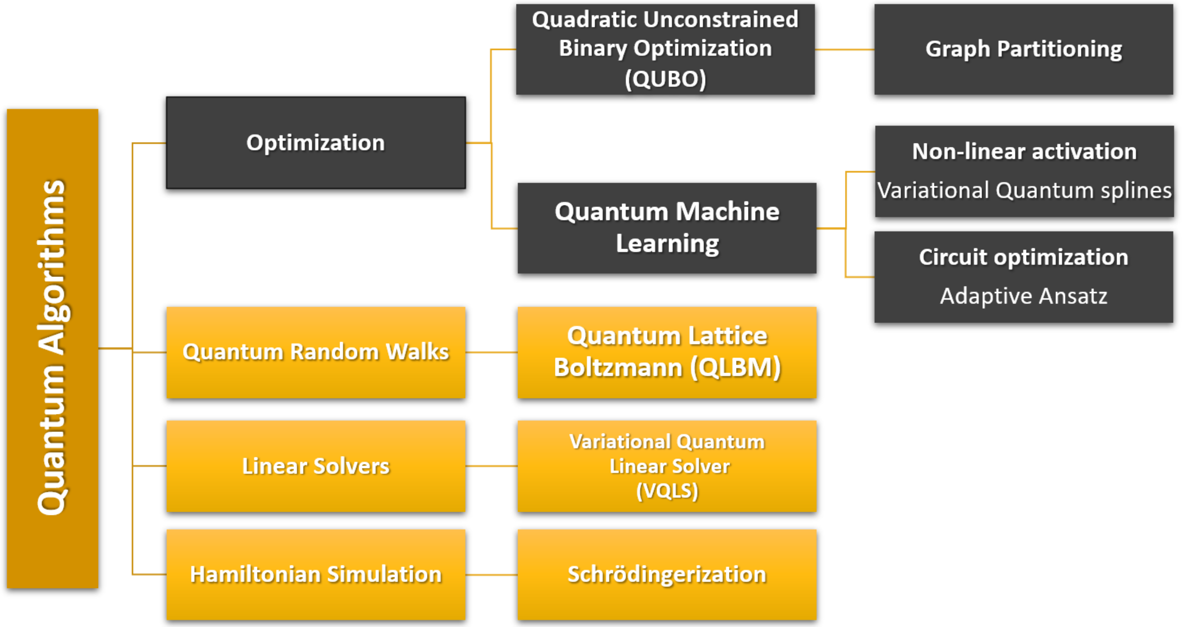


The CUDA Quantum Stack

Platform for unified quantum-classical accelerated computing



Quantum Computing CAE Simulation



Variational Quantum Algorithms: 0-3 Yrs.
- Hybrid quantum-classical linear solvers
- Paves the way to HHL

Quantum Machine Learning: 0-3 Yrs.
- Use ML to map the problem from classical to quantum space
- ML for efficient allocation of compute resources (# qubits)

Graph Partitioning: 0-3 Yrs.
- Min cut NP complete graph problem
- Faster matrix reordering, bottleneck step in MAPDL/LSDyna

Quantum Optimization: 3-6 Yrs.
- Use of quantum annealing (D-Wave)
- Leap from hybrid (as in VQAs) to purely quantum optimization

Quantum Lattice Boltzmann: 3-6 Yrs.
- Quantum analog of classical "random walks"
- Is quantum native, no artificial morphing into optimization needed.

Harrow-Hassidim-Lloyd (HHL): 6-10 Yrs.
- Solve Ax = b in CAE simulation
- Challenge: hardware insufficient (is small, noisy)

Company Partnerships

IBM, NVIDIA, IONQ

Startup Partnerships

QUANSCIENT, PsiQuantum, PASQAL, QUANTINUUM

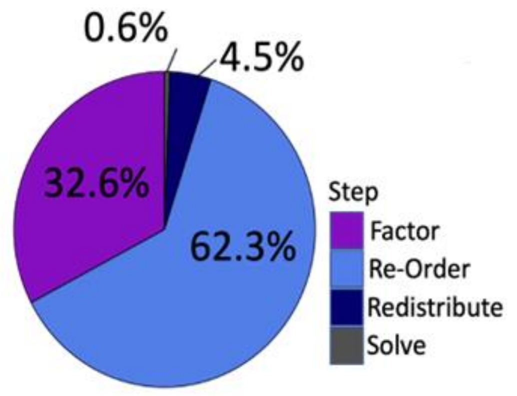
Academic Partnerships

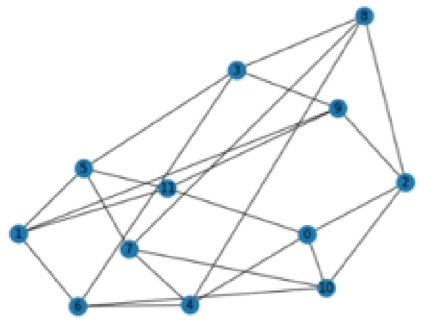
Carnegie Mellon University, Technion Israel Institute of Technology, Indian Institute of Technology Madras

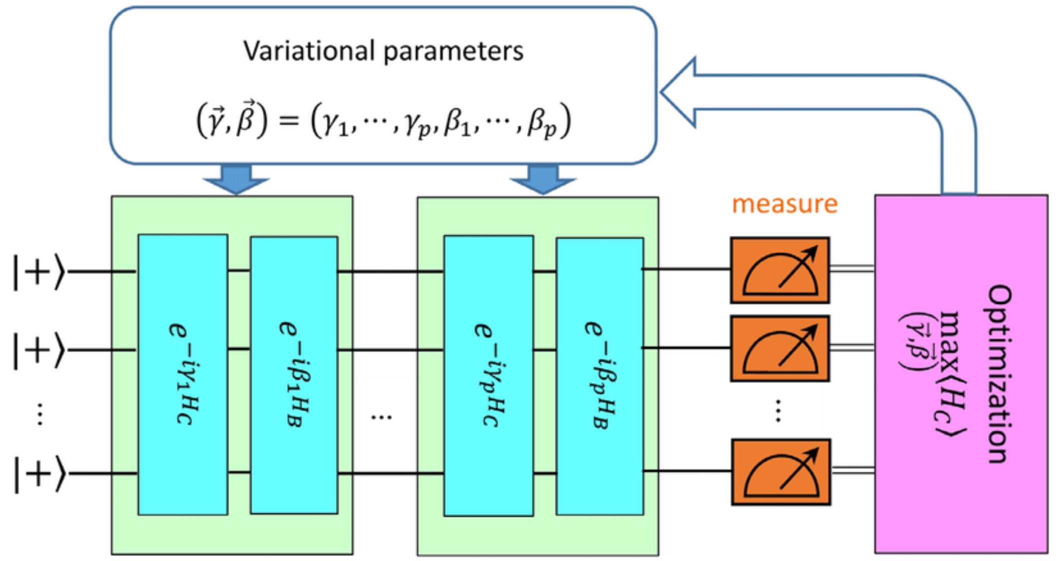
Graph Partitioning

Graph partitioning

- Accelerating graph partitioning would allow us to speed up two of our flagship solvers, MAPDL and LS-DYNA
- Example Rolls Royce Engine run time for LS-DYNA shows Matrix Re-Ordering step takes 62% of time
- Re-Ordering step can be solved by Nested Dissection, which is form of graph partitioning
- Graph partitioning is an NP-complete problem and as such is exponentially hard to solve by classical means.
- Combine a Quantum Computer (solve the energy minimization problem) with a Classical Computer (to optimize the parameters of quantum computer)
- Various quantum algorithms to solve them: Variational Quantum Eigensolver (VQE), Quantum Approximate Optimization Algorithm (QAOA),…
- Challenges: (1) one qubit per node does not scale (2) explore encodings of multiple nodes to one qubit (3) increased depth with encoding (4) Noisy qubits and error correction
- Partnership with DWave, Pascal, IBM, CMU

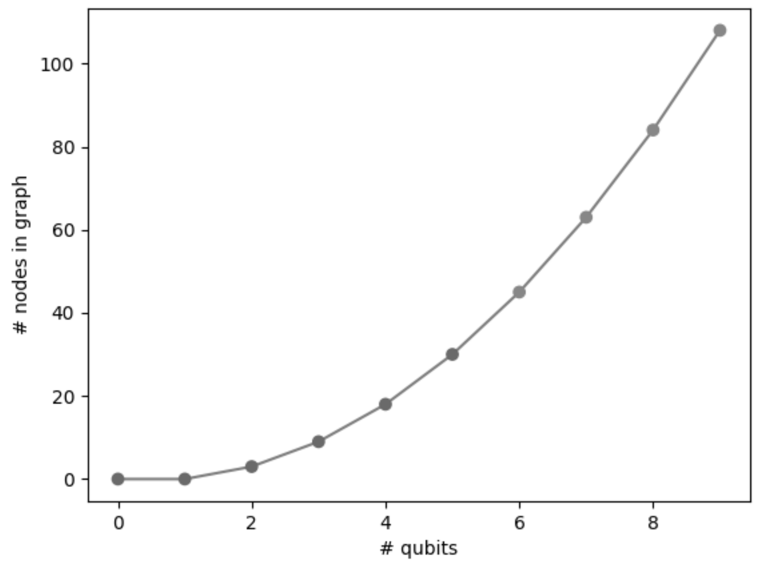






Graph partitioning

- QITE approach with IonQ: n node => n qubits
  - Upto 32 qubits on IonQ hardware
  - Smooth convergence
- Ansys Research: 1.5 * n * (n-1) nodes => n qubits
  - Encoding binary variables in correlations between qubits
  - Can handle 100 nodes problems with 9 qubits
- Current hurdles:
  - Quantum solutions' approx. ratios get relatively worse than METIS' as graph sizes increase.
  - Number of iterations required for variational algorithms render them unappealing in terms of time and cost.
- Current idea being implemented:
  - Train simplified circuit classically before optimizing complicated circuit. This may help address either or both hurdles mentioned.

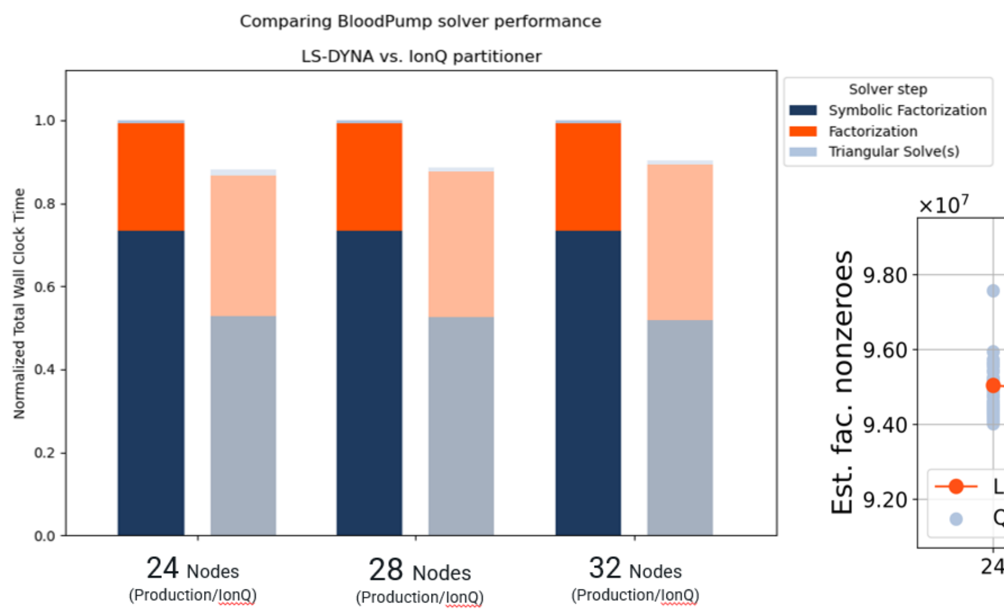


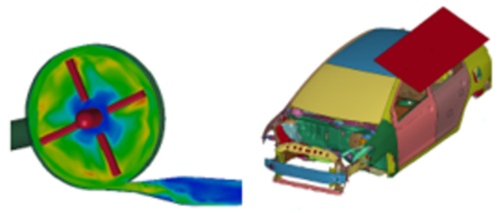
100-node problem instances, 9 qubit, quantum algorithm outperforms a classical optimizer

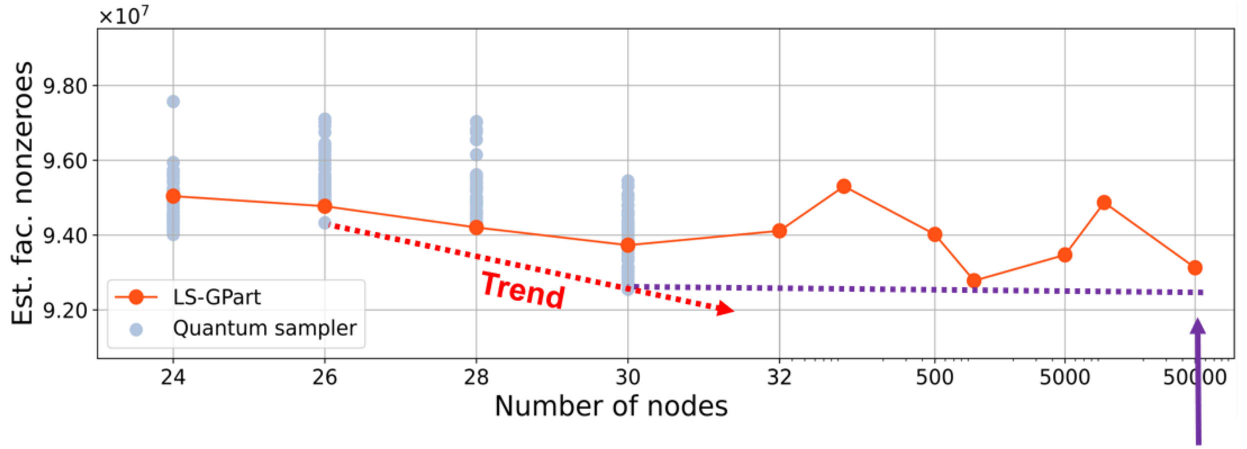
Graph Partitioning on IONQ Hardware

Quantum graph partitioning for LS-DYNA acceleration

Comparison of wall-clock run-time between LS-DYNA full-scale model, coarse model, and quantum-processed coarse model




Blood Pump (2.6M vertices) Crushed Toyota (600K vertices)



Sampled GPP solutions from QITE for sub-problem size on 30 nodes/qubits improves over LS-DYNA sub-problem size 50k nodes

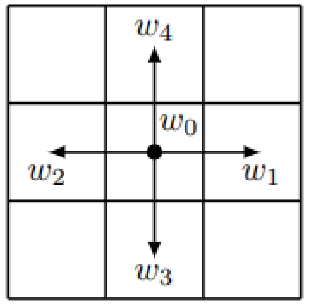
- Model: 2.6M vertices and 40.6M edges
  Baseline: coarsened LS-Part to production
  System: 4 Intel Xeon Gold 6242 CPUs, 64 cores in total; and 1.5TB of memory

Quantum accelerated workflow resulting with meaningful wall-clock improvement of LS-DYNA simulation time

Lattice Boltzmann for CFD Simulation

Quantum Lattice Boltzmann Method for Fluids

- Lattice Boltzmann approach to CFD
- Consider a 1D, 2D, or 3D Cartesian lattice of points, and solve for **ensemble of particles** flowing through space (Advection-Diffusion equations)
- **Two main steps**: (1) Streaming (or propagation of particles)
  (2) Collision (inter-particle interaction from neighboring sites)
- LBM can model viscosity and fluids and result in Navier Stokes equations
- Industrial strength Lattice Boltzmann methods applied to solving turbulent flow for airplanes and external airflow for automobiles
- For 500 million to 2 billion lattice points, can take days or weeks to run (with quantum can run in seconds)
- Lattice Boltzmann method is **amenable to quantum computing**. Streaming step can be cast as a quantum random walk. Newer, more efficient collision operators for quantum circuits regularly published in literature.
- Academic 1D and 2D quantum formulations exist, working on inhouse modular implementations. Two streams for testing: scalable simulation with HPC, and quantum hardware runs.
- Partnership with Quanscient, Technion



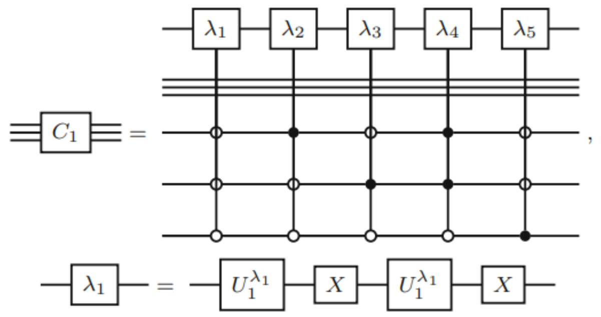
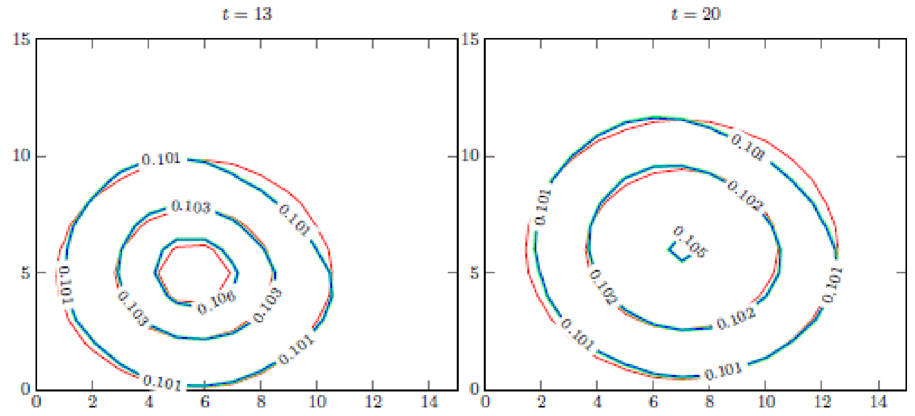


Fig. 7 Quantum circuit for implementing the operator C_1 in case of the D2Q5 LBM



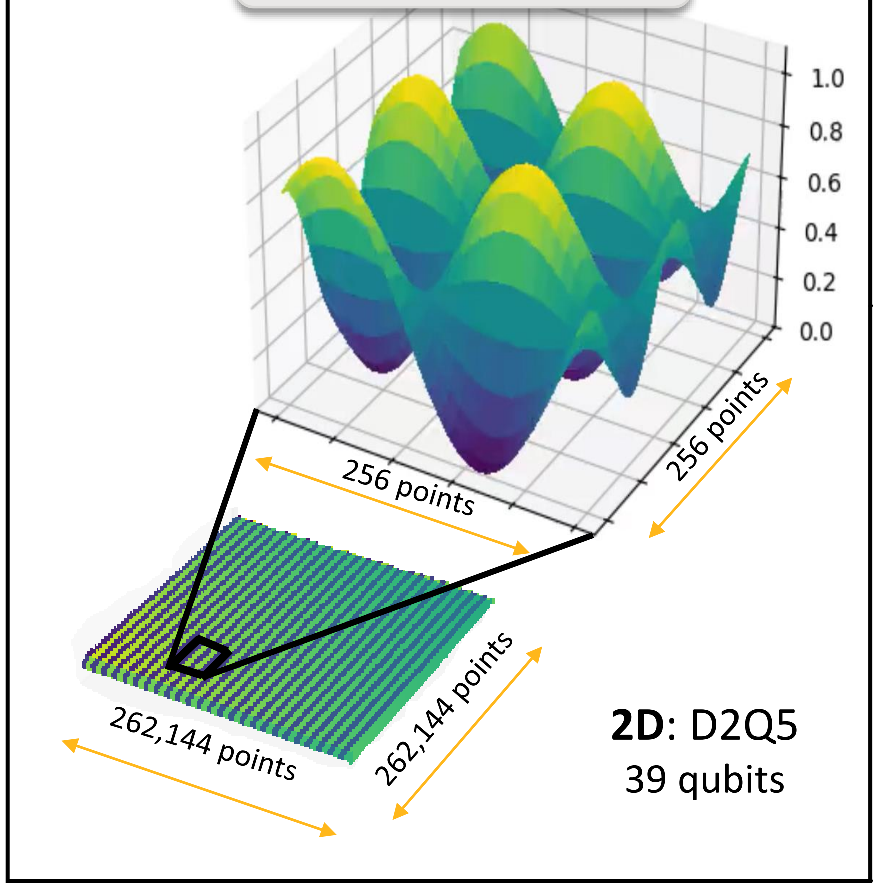
Comparison of the numerical results obtained by the quantum algorithm, the classical FORTRAN code, and the analytical solution of the 2D ADE (Budinski, 2021)

QLBM on CUDA-Q emulator

Ansys x Nvidia

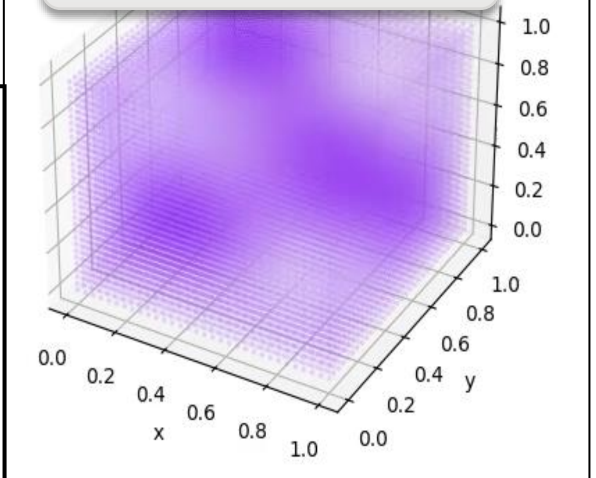
- Scaled to 39 qubits, 68 bil. gridpoints
- Non uniform velocity fields: shear and swirl
- Simulated on upto 1464 H100 GPUs

Advection diffusion of 2D sinusoid in uniform flow

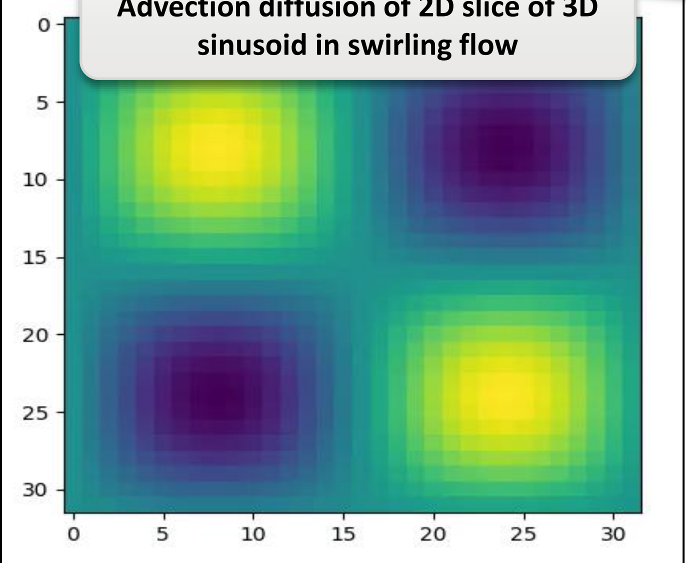


2D: D2Q5
39 qubits

Advection diffusion of 3D sinusoid in swirling flow

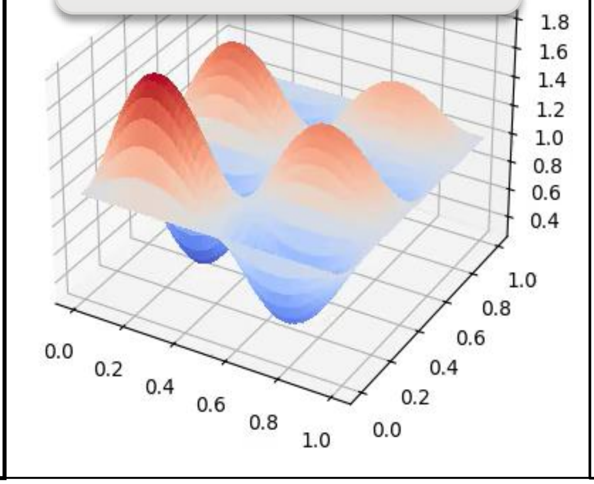


Advection diffusion of 2D slice of 3D sinusoid in swirling flow

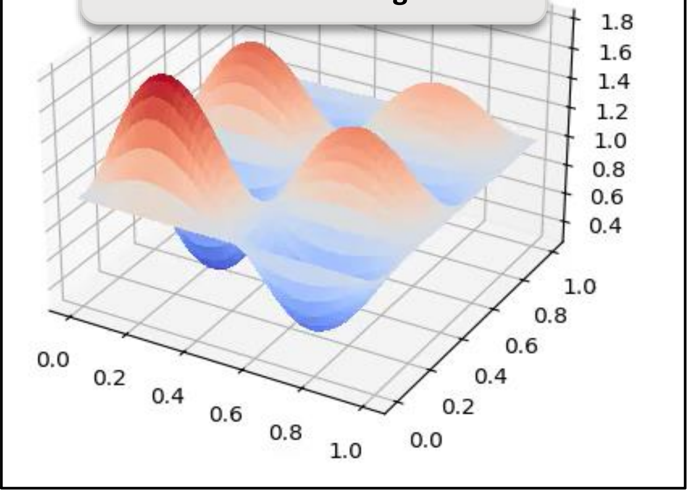


3D: D3Q7
32x32x32 lattice
18 qubits

Advection diffusion of 2D sinusoid in shear flow



Advection diffusion of 2D sinusoid in swirling flow

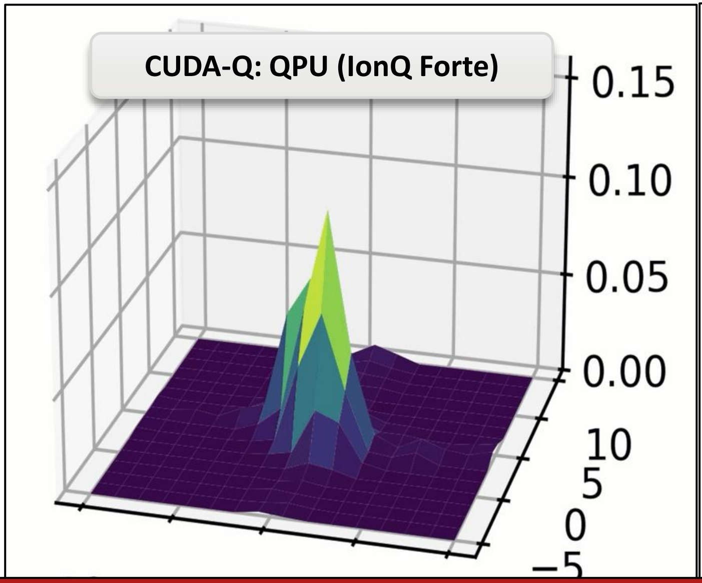


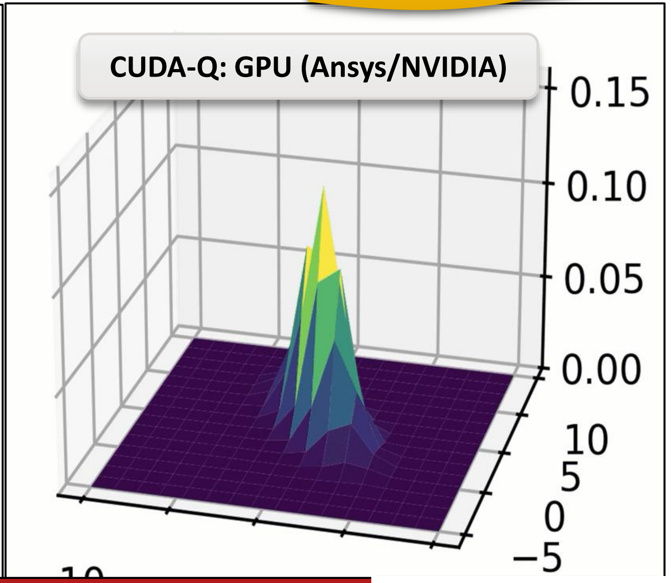
2D: D2Q5
32768x32768 lattice
33 qubits

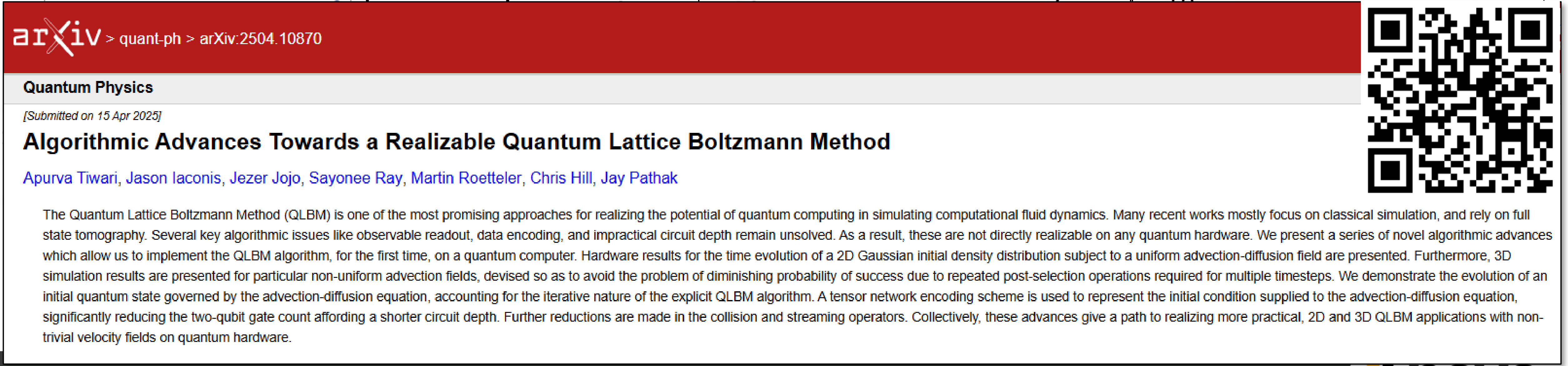
QLBM on IonQ hardware - Uniform advection diffusion of 2D gaussian

Ansys x IonQ

- **Identify observables:** scalar: flux. system: lift/drag coefficients.
- **Multi timesteps:** No state tomography
- **Efficient data loading** using MPS(matrix product states)
- **Problem size:** 16x16 grid, 10 time steps
- **19 qubits, 270 2q-gates with error mitigation**
- **Circuit optimization**
  - Reduction of 2 qubit gates
  - One-hot encoding of directional qubits in LBM







arXiv > quant-ph > arXiv:2504.10870

Quantum Physics

[Submitted on 15 Apr 2025]

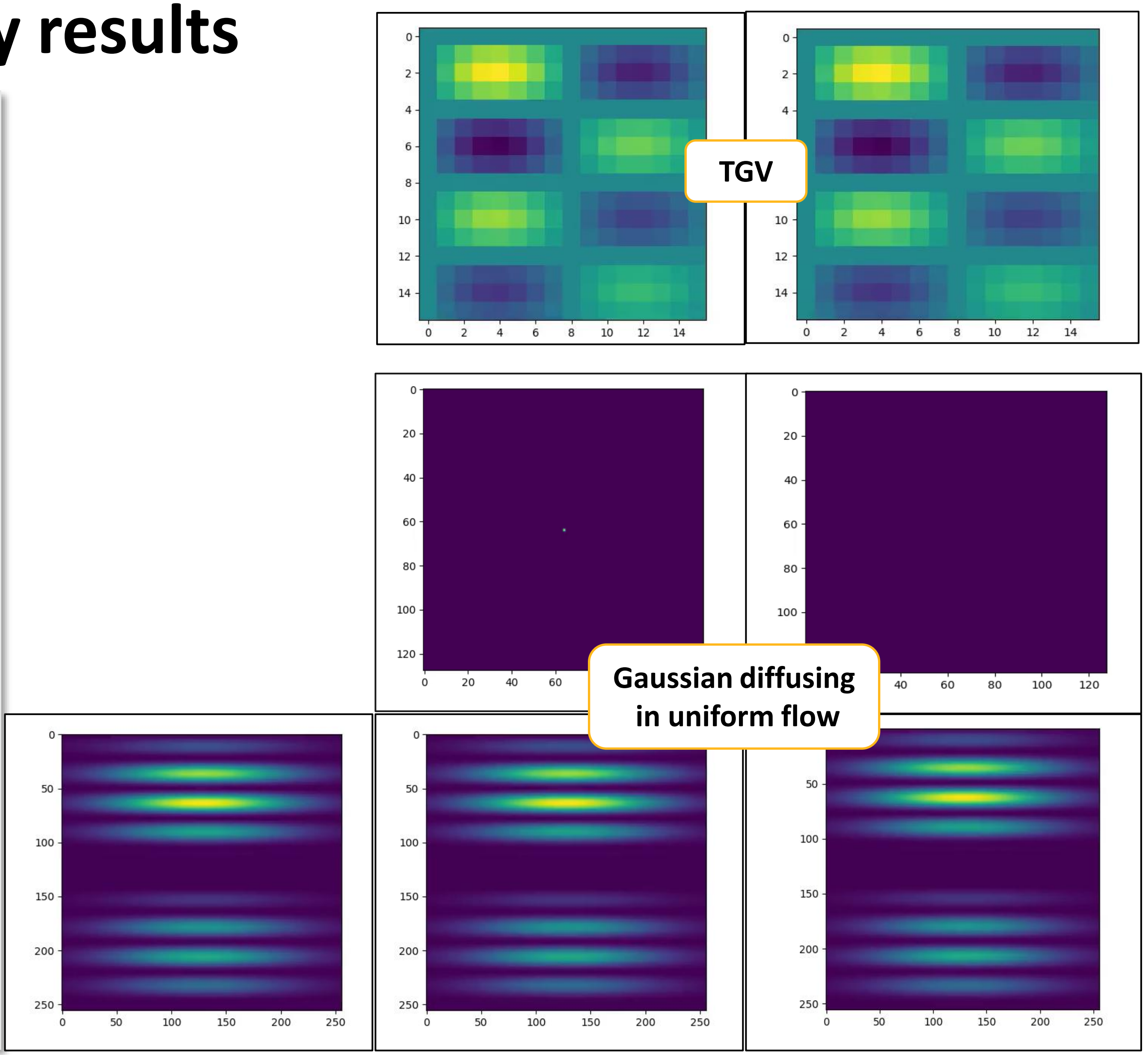
Algorithmic Advances Towards a Realizable Quantum Lattice Boltzmann Method

Apurva Tiwari, Jason Iaconis, Jezer Jojo, Sayonee Ray, Martin Roetteler, Chris Hill, Jay Pathak

The Quantum Lattice Boltzmann Method (QLBM) is one of the most promising approaches for realizing the potential of quantum computing in simulating computational fluid dynamics. Many recent works mostly focus on classical simulation, and rely on full state tomography. Several key algorithmic issues like observable readout, data encoding, and impractical circuit depth remain unsolved. As a result, these are not directly realizable on any quantum hardware. We present a series of novel algorithmic advances which allow us to implement the QLBM algorithm, for the first time, on a quantum computer. Hardware results for the time evolution of a 2D Gaussian initial density distribution subject to a uniform advection-diffusion field are presented. Furthermore, 3D simulation results are presented for particular non-uniform advection fields, devised so as to avoid the problem of diminishing probability of success due to repeated post-selection operations required for multiple timesteps. We demonstrate the evolution of an initial quantum state governed by the advection-diffusion equation, accounting for the iterative nature of the explicit QLBM algorithm. A tensor network encoding scheme is used to represent the initial condition supplied to the advection-diffusion equation, significantly reducing the two-qubit gate count affording a shorter circuit depth. Further reductions are made in the collision and streaming operators. Collectively, these advances give a path to realizing more practical, 2D and 3D QLBM applications with non-trivial velocity fields on quantum hardware.

3D; Arbitrary tau preliminary results

- **3D**
  - Advection-diffusion using QLBM, generalized for **any divergence-free velocity field**
  - QLBM D3Q27 model implemented on CUDA-Q.
  - Implementation is for uniform velocity (lower panels) and Taylor-Green vortex (upper panel)
  - Just like the 2D case,
    certain non-trivial velocity fields can be implemented.
  - Taylor-Green vortex uses methods in previous literature. Meant only for emulation, runtime issues if run on actual hardware (even ideal noise-free).
- **Arbitrary tau = Arbitrary diffusion constant**
  - 'Tau' is a parameter related to the viscosity of the fluid.
  - All existing literature sets tau to be 1.
    We produced a novel method to work with any value of tau.

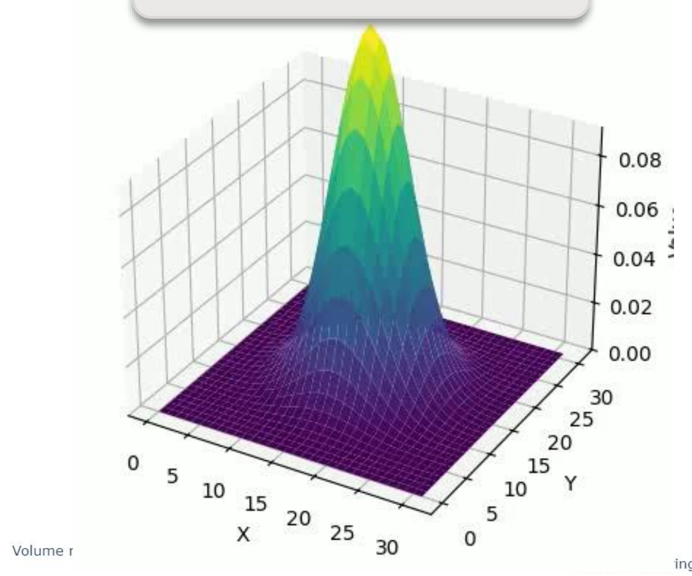


Hamiltonian Simulation

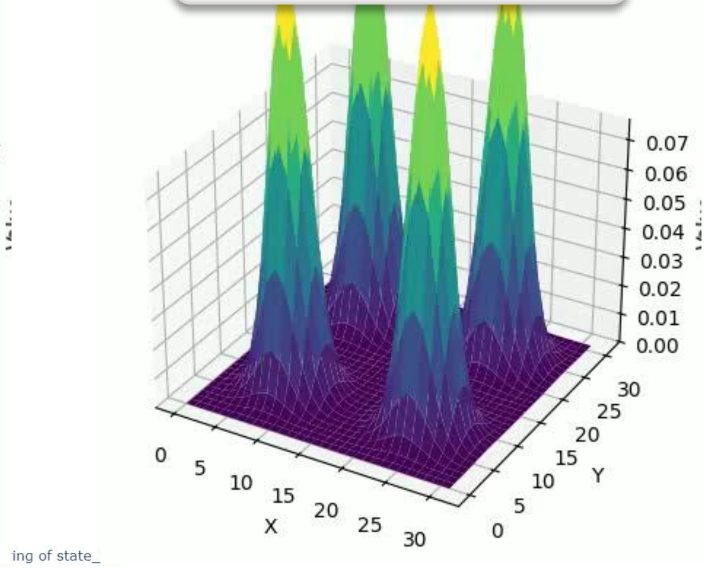
Hamiltonian Simulation on CUDA-Q Emulator

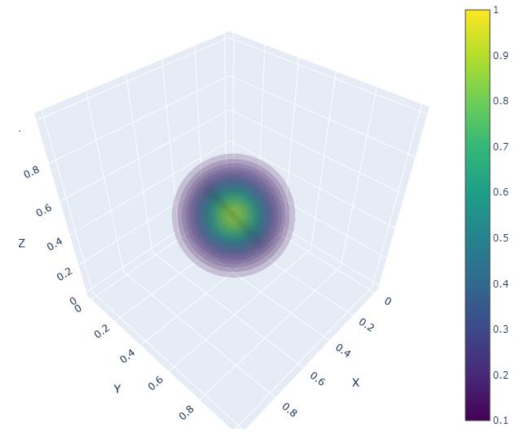
- Cast governing equations (e.g. Navier-Stokes) into Schrödinger. Let qubits evolve. Map back to target physics.
- Simulation on Cuda-Q
- Rotational velocity fields
- Circuit depth polynomial w.r.t qubits
- Advection and Diffusion scaled to **27 qubits**
- **Shear, Rotation and polynomial velocity fields**
- Energy and Flux measurement with an **observable –** much efficient than reading full solution

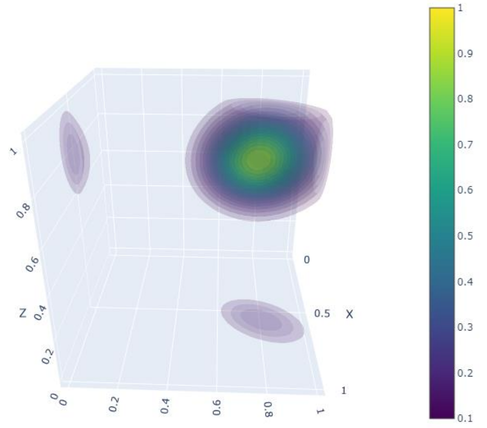
Advection diffusion of 2D sinusoid in uniform flow



Advection diffusion of 2D sinusoid in rotational flow



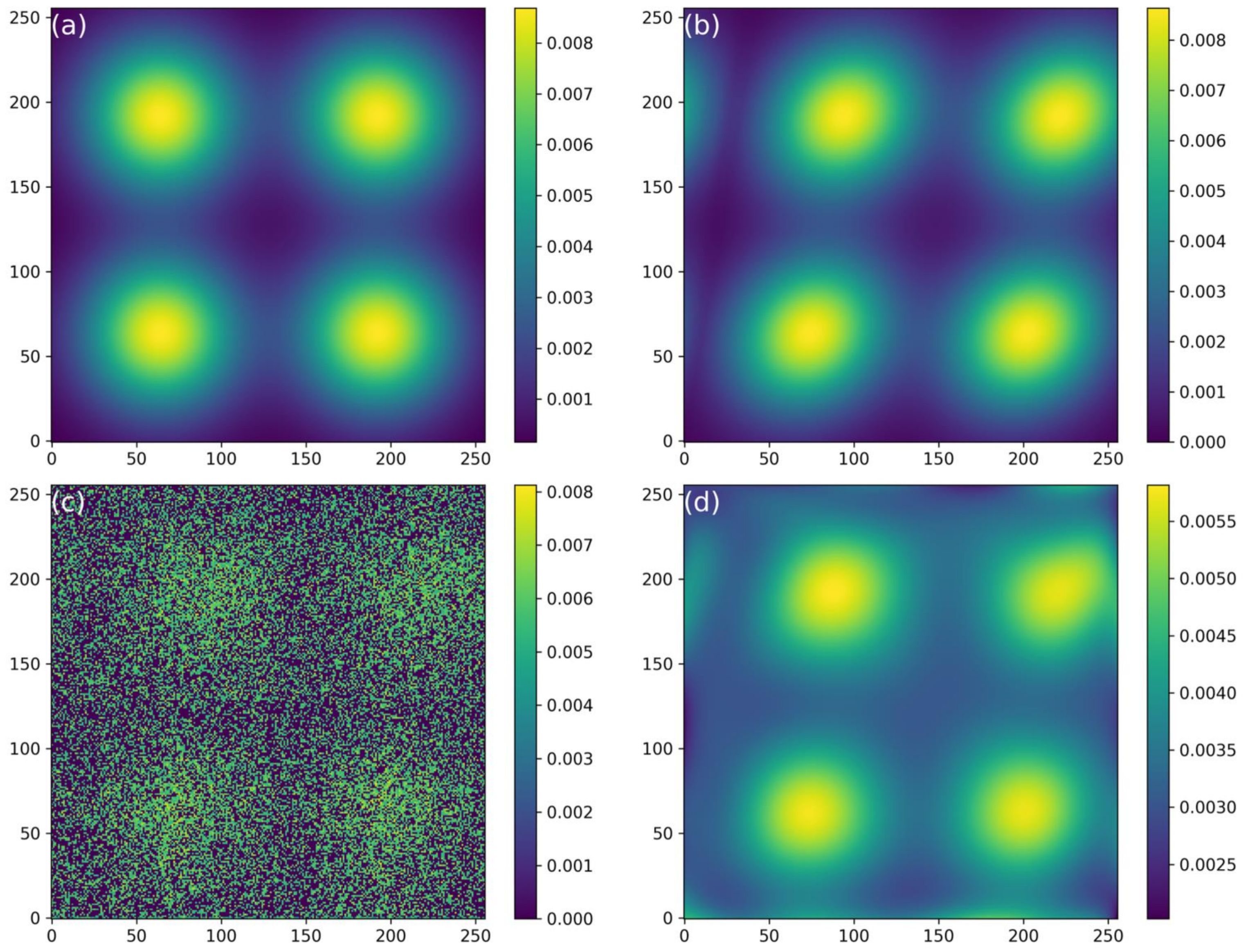




3D Shear, 8+8+8+3= 27 qubits

Hardware run (8+8 qubits) circuit depth ~ 400 (compared to 100K)

T=40, dt=1.0



Discussion and Next Steps

Quantum Computing is the next technological disruption after AI/ML

Ansys has started investing in topic

- Quantum Computers to Accelerate Simulation
  - Graph Partitioning
  - Quantum Lattice Boltzmann method for Fluids
  - Hamiltonian Simulation

Ansys