



A Quantum Solver for Multidimensional Partial Differential Equations: Practical Case Studies

Manu Chaudhary¹, Kareem El-Araby², Alvir Nobel², Ishraq Islam², Manish Singh², Sunday Ogundele², Kieran Egan², Sneha Thomas², Vincent Vordtriede², Devon Bontrager², Serom Kim², and Esam El-Araby²

mchaud4@ilstu.edu, {kareem_el_araby, islam.alvir, ishraq, manish.singh, sunday, kieran.fo.egan, snehathomas, vvor3, devonbontrager, serom.kim, and esam}@ku.edu

¹Electrical Engineering, Illinois State University (ISU), ²Electrical Engineering and Computer Science (EECS), University of Kansas (KU)



Introduction and Motivation

- Quantum computing (QC) has the potential to significantly transform computational problem-solving.
- Classical methods for solving complex and multidimensional partial differential equations (PDEs) suffer from an exponential growth in computational complexity known as the curse of dimensionality.
- In contrast, quantum computing (QC) inherently exploits superposition and entanglement to efficiently represent and simultaneously manipulate large-scale discretized PDEs, using exponentially fewer resources compared to classical systems.
- There are currently many quantum techniques available for solving PDEs, which are mainly based on variational quantum circuits and the Harrow-Hassidim-Lloyd (HHL) algorithm.
- However, the existing quantum PDE solvers, particularly those based on variational quantum eigensolver (VQE) techniques, suffer from several limitations, including low accuracy, high execution times, and low scalability on quantum simulators as well as on noisy intermediate-scale quantum (NISQ) devices.
- The HHL algorithm requires ancilla qubits for solution extraction and multiple qubits for eigenvalue precision, increasing resource demands. It also needs a sparse, well-conditioned Hermitian matrix to achieve a logarithmic gate complexity. Furthermore, HHL's reliance on quantum phase estimation (QPE), which is highly sensitive to noise, makes it unsuitable for noisy-intermediate scale quantum (NISQ) devices, necessitating error-corrected quantum hardware.
- In this work, we propose an efficient, accurate, and scalable quantum algorithm for solving multidimensional PDEs. We present two variants of our algorithm: the first leverages finite-difference method (FDM), classical-to-quantum (C2Q) encoding, and numerical instantiation, while the second employs FDM, C2Q, and column-by-column decomposition (CCD). Both variants are designed to enhance accuracy and scalability while reducing total execution times.
- We validated our proposed algorithm using several case studies including multidimensional Poisson, Heat, Black-Scholes, and Navier-Stokes equations.
- Our experimental work demonstrated favorable results in terms of accuracy, scalability, and execution time compared to quantum variational solvers on noise-free/noisy simulators, hardware emulators, and real quantum hardware from IBM.

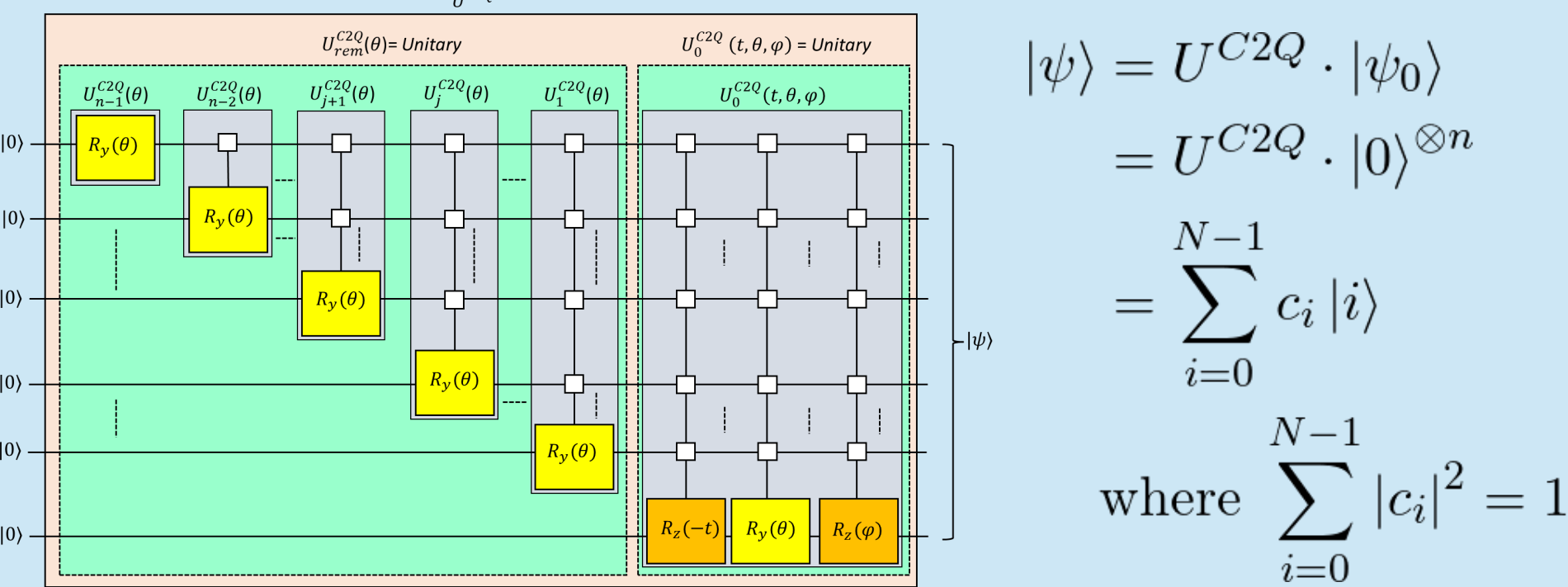
Background

- Finite Difference Method (FDM)**
- Numerical technique to solve PDEs by approximating them with difference equations.
- Converts a continuous domain problem into a finite set of grid points.

$$A\vec{u} = \vec{b}$$

Fundamentals of Quantum Computing

- Quantum computers utilize superposition and entanglement.
 - Quantum bit or qubit is the fundamental unit of quantum computation.
 - For an n-qubit system, the quantum state vector is represented as shown:
- $$|\psi\rangle = \sum_{j=0}^{N-1} c_j |j\rangle, \text{ where } \sum_{j=0}^{N-1} |c_j|^2 = 1, \text{ and } 0 \leq j < N$$
- Quantum gates are unitary operations which can change the state of a qubit.
 - Classical-to-quantum encoding (C2Q)**
 - C2Q is a depth-optimized amplitude encoding technique [1] to encode classical data into quantum domain.
 - Encodes normalized classical data vector into quantum domain.
 - Generates shallower circuit by using 50% fewer gates compared to nearest alternative [2].
 - Suitable for NISQ devices because shallow circuits are more resistant to decoherence noise.



Singular Value Decomposition (SVD)

- Decompose a matrix into two orthogonal matrices and a diagonal matrix.
- $$A = W\Sigma V^T$$

Polar Decomposition

- Decompose a matrix into two unitary matrices and a positive semi-definite symmetric matrix.
- $$A = PU$$

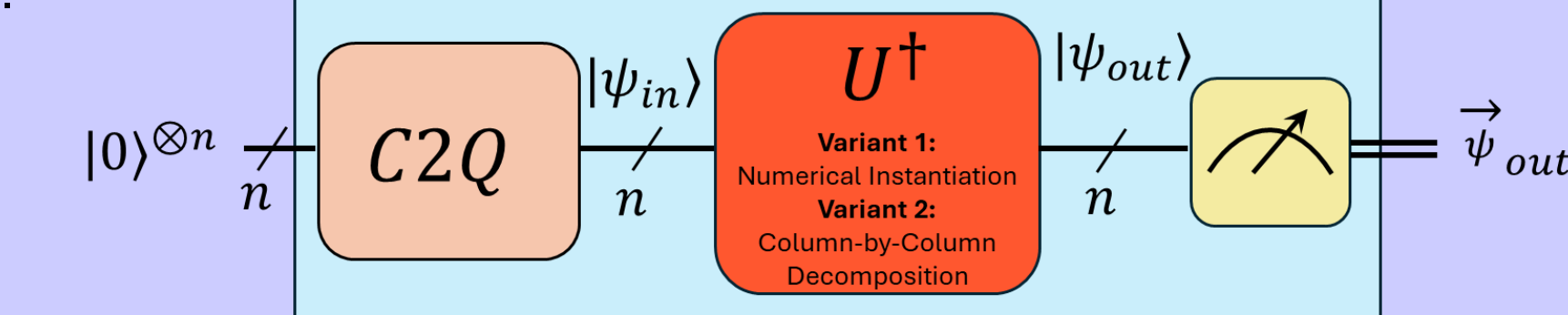
Related Work

- Performance study of VQAs for solving the Poisson equation on a quantum computer [6]
- Uses variational quantum linear solver (VQLS) for solving Poisson equations
- Poor hardware results, low accuracy and long execution times
- Variational quantum algorithm based on the minimum potential energy for solving the Poisson equation [7]
- Solves the Poisson equation by formulating a minimum potential energy-based cost function
- VQE-based approach for solving PDEs but it exhibits low accuracy and long execution times on NISQ devices
- Quantum algorithm for systems of linear equations with exponentially improved dependence on precision [8]
- Requires significantly less time and has a better precision than HHL algorithm
- Uses Fourier or Chebyshev series
- Not suitable for NISQ devices because of their high error rates, limited coherence, and complex operations
- Hybrid classical-quantum approach to solve heat equation using quantum annealers [9]
- Linear system is encoded as a quadratic unconstrained binary optimization (QUBO) problem
- Solves the steady-state 2D heat equation using D-Wave systems
- Time dependent heat analysis not shown, and the approach suffers from scalability issues
- A quantum approach for exploring the numerical results of the heat equation [10]
- Solves 1D heat equation using Trotter Suzuki decomposition
- Suffers from high error rate and limited scalability

Proposed Methodology

Proposed PDE Solver for Multidimensional PDEs

- The proposed PDE solver algorithm has two variants.
- Variant 1 uses classical-to-quantum (C2Q) encoding, finite difference method (FDM), and numerical instantiation [3], whereas Variant 2 uses C2Q, FDM, and column-by-column decomposition (CCD) [4].



- Using FDM, the partial differential equation (PDE) is converted into a system of linear equations.
- $$A\vec{u} = \vec{b}$$

SVD and polar decomposition has been used to find the value of P^{-1} and U^T as shown below: $A = PU = W\Sigma V^T$, where $P = W\Sigma W^T$, $U = WV^T$, $P^{-1} = W\Sigma^{-1}W^T$, $U^T = V^T W^T$

System matrix A is preprocessed using polar decomposition, rearranged, and normalized as shown:

$$(PU)\vec{u} = \vec{b} \Rightarrow U\vec{u} = P^{-1}\vec{b} \Rightarrow \vec{u} = U^T P^{-1}\vec{b} \Rightarrow \vec{u} = U^T \left(\frac{P^{-1}\vec{b}}{\|P^{-1}\vec{b}\|} \right)$$

$$|\psi_{out}\rangle = U^T \cdot |\psi_{in}\rangle, \text{ where } |\psi_{out}\rangle = \frac{\vec{u}}{\|P^{-1}\vec{b}\|}, \text{ and } |\psi_{in}\rangle = \frac{P^{-1}\vec{b}}{\|P^{-1}\vec{b}\|}$$

- The unitary matrix U^T is converted into a quantum circuit using numerical instantiation as implemented in BQSKIT [3] in Variant 1, and using CCD algorithm [4] as implemented in Qiskit [5] in Variant 2.
- $\vec{\psi}_{out} \approx U^T |\psi_{in}\rangle$, $\vec{u} = \vec{\psi}_{out} \|W\Sigma^{-1}W^T\vec{b}\|$, $|\psi_{in}\rangle = \|W\Sigma^{-1}W^T\vec{b}\|$

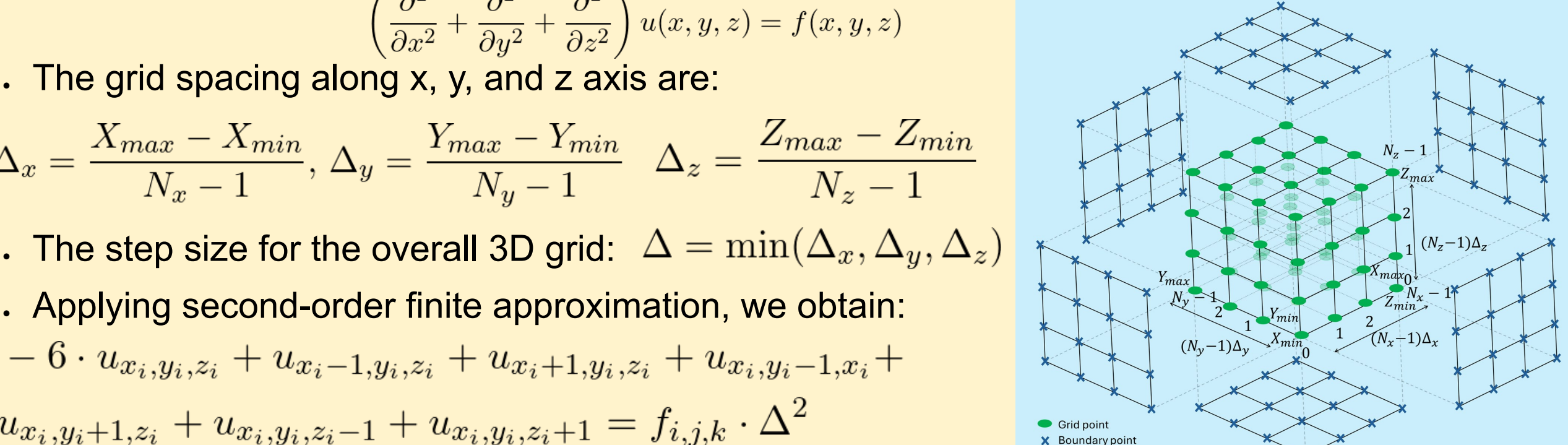
- By measuring $|\psi_{out}\rangle$ for large number of shots/samples, we recover the classical vector $\vec{\psi}_{out}$ and finally obtain the unknown vector \vec{u} .

Case Study 1: Multidimensional Poisson Equation

- A p-dimensional Poisson equation can be represented as:
- $$\nabla^2 u = \sum_{i=1}^p \frac{\partial^2 u}{\partial s_i^2} = f(s_1, s_2, \dots, s_p)$$

where u is unknown function, f is Source function, and ∇^2 is Laplacian operator.

- A 3D Poisson equation can be represented as:
- $$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) u(x, y, z) = f(x, y, z)$$

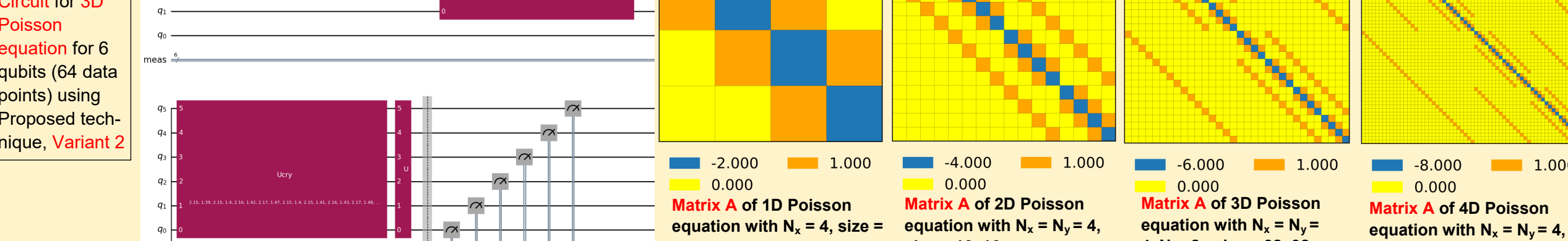


The grid spacing along x, y, and z axis are:

$$\Delta_x = \frac{X_{max} - X_{min}}{N_x - 1}, \Delta_y = \frac{Y_{max} - Y_{min}}{N_y - 1}, \Delta_z = \frac{Z_{max} - Z_{min}}{N_z - 1}$$

- The step size for the overall 3D grid: $\Delta = \min(\Delta_x, \Delta_y, \Delta_z)$
 - Applying second-order finite approximation, we obtain:
- $$-6 \cdot u_{x_i, y_j, z_k} + u_{x_i+1, y_j, z_k} + u_{x_i-1, y_j, z_k} + u_{x_i, y_j+1, z_k} + u_{x_i, y_j-1, z_k} + u_{x_i, y_j, z_k+1} + u_{x_i, y_j, z_k-1} = f_{i,j,k}$$

It can be further reduced to:

$$A\vec{u} = \vec{b}$$


Case Study 2: Multidimensional Heat Equation

- A p-dimensional space-time, (p-1)D space + time, Heat equation can be represented as:
- $$\frac{\partial u(s_1, s_2, \dots, s_{p-1}, t)}{\partial t} = \nabla^2 u = \sum_{i=1}^{p-1} \frac{\partial^2 u(s_1, s_2, \dots, s_{p-1}, t)}{\partial s_i^2}$$

- A 4D space-time heat equation can be represented as:
- $$\frac{\partial u(x, y, z, t)}{\partial t} = \alpha_x \frac{\partial^2 u(x, y, z, t)}{\partial x^2} + \alpha_y \frac{\partial^2 u(x, y, z, t)}{\partial y^2} + \alpha_z \frac{\partial^2 u(x, y, z, t)}{\partial z^2}$$

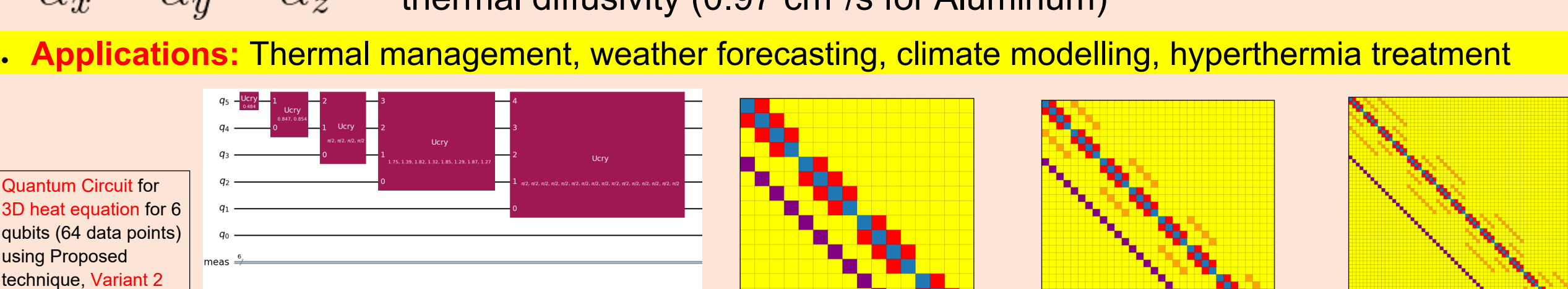
Applying second order finite approximation:

$$-(1 + 2(\lambda_x + \lambda_y + \lambda_z))u_{i,j,k,t} + \lambda_x u_{i+1,j,k,t} + \lambda_x u_{i-1,j,k,t} + \lambda_y u_{i,j+1,k,t} + \lambda_y u_{i,j-1,k,t} + \lambda_z u_{i,j,k+1,t} + \lambda_z u_{i,j,k-1,t} = 0$$

$$\Delta_x = \frac{X_{max} - X_{min}}{N_x - 1}, \Delta_y = \frac{Y_{max} - Y_{min}}{N_y - 1}, \Delta_z = \frac{Z_{max} - Z_{min}}{N_z - 1}, \lambda_x = \frac{\Delta_x}{\Delta t} \alpha_x, \lambda_y = \frac{\Delta_y}{\Delta t} \alpha_y, \lambda_z = \frac{\Delta_z}{\Delta t} \alpha_z$$

$\alpha_x = \alpha_y = \alpha_z = \text{thermal diffusivity } (0.97 \text{ cm}^2/\text{s for Aluminum})$

- Applications: Thermal management, weather forecasting, climate modelling, hyperthermia treatment



Case Study 3: Black-Scholes Equation

The Black-Scholes model is a European-style option pricing model that can be exercised only at its expiration date.

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0$$

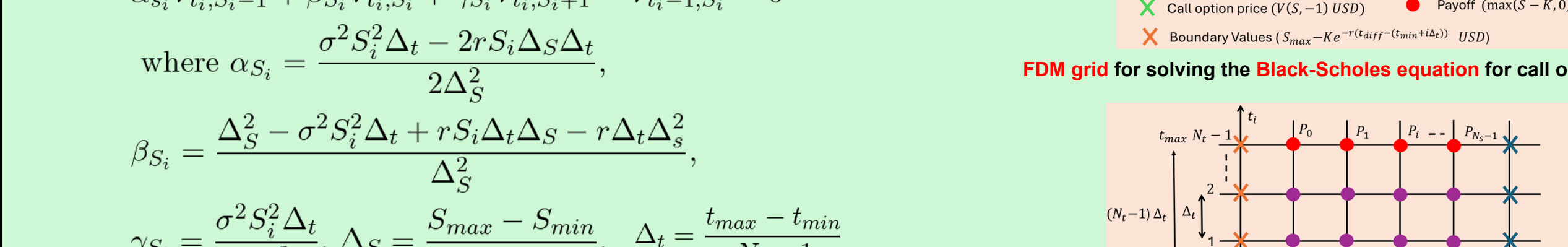
Here, S is the stock price, V is option price, σ is volatility of the asset, r is risk-free interest rate, and K is strike price.

Call (Buy) Option

- A call option is contract that gives the buyer right, but not the obligation to purchase a specific stock at the strike price K on a pre-determined expiration date.
- Applying FDM, the discretized difference equation obtained is shown below:

$$\alpha_n V_{i,n} - \beta_{i,n} V_{i-1,n} + \gamma_{i,n} V_{i+1,n} - V_{i,n} - \delta_{i,n} = 0$$

where $\alpha_n = \frac{\sigma^2 S_{i,n}^2 \Delta t}{2\Delta S^2}$, $\beta_n = \frac{\Delta S^2}{2\Delta S^2} - \sigma^2 S_{i,n}^2 \Delta t + rS_{i,n} \Delta t - r\Delta t \Delta S_{i,n}$, $\gamma_n = \frac{\sigma^2 S_{i,n}^2 \Delta t}{2\Delta S^2}$, $\delta_n = \frac{S_{max} - S_{min}}{N_S - 1}$, $\Delta t = \frac{t_{max} - t_{min}}{N_t - 1}$

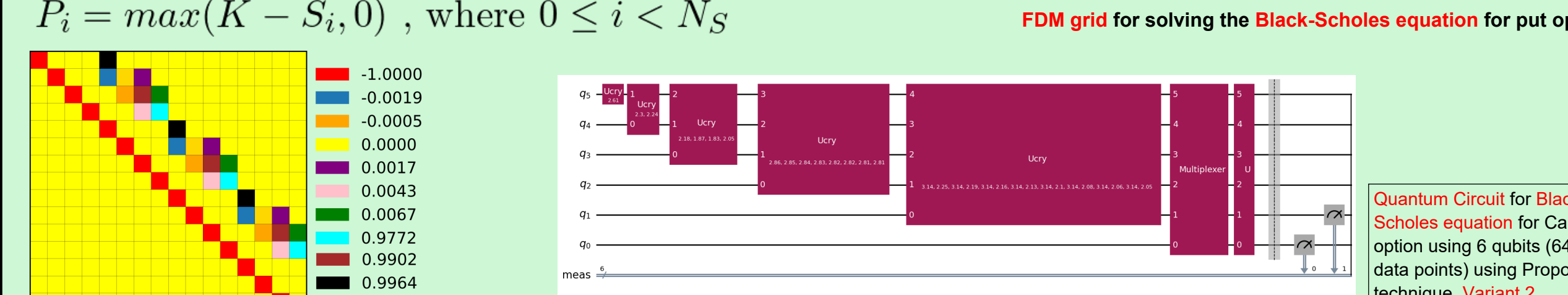


The payoff for call option is represented as:

$$P_i = \max(S_i - K, 0), \text{ where } 0 \leq i < N_S$$

Put (Sell) Option

- The payoff value for put option is represented as:
- $$P_i = \max(K - S_i, 0), \text{ where } 0 \leq i < N_S$$



Applications: Option pricing, risk management, financial decision making

Case Study 4: Navier-Stokes Equation

- The Navier-Stokes equation governs the momentum transport in a Newtonian fluid of constant density ρ and kinematic viscosity ν as shown:

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \vec{u} + \vec{f}$$

Here, \vec{u} is unknown velocity and \vec{f} is the body force, and P is pressure.

2D (1 space + time) Navier-Stokes Equation

- A 2D space-time equation can be represented as shown below. Here, the flow is in the x-direction and velocity has only x-component.

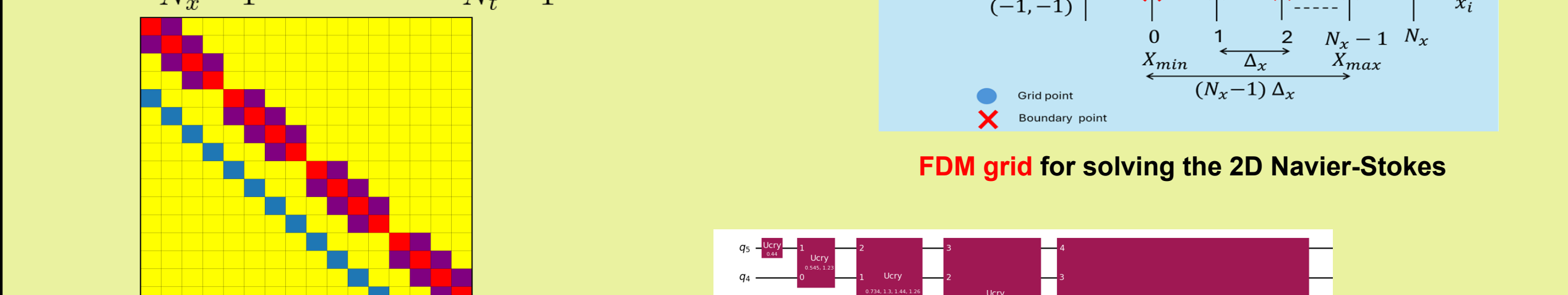
$$u \frac{\partial u}{\partial t} - \nu \frac{\partial^2 u}{\partial x^2} = -\frac{1}{\rho} \frac{\partial P}{\partial x} + f_x$$

Considering $P = P_{in} - G \cdot x$ and applying second order finite approximation, we get:

$$\alpha u_{i,j,t} - \beta u_{i-1,j,t} + \gamma u_{i+1,j,t} - u_{i,j,t} = \lambda \left(\frac{G}{\rho} + f_x \right)$$

where $\alpha = \frac{\Delta t}{\nu} \Delta x^2$, $\beta = \frac{\Delta t}{\nu} \Delta x^2$, $\gamma = \frac{\Delta t}{\nu} \Delta x^2$, $\lambda = \frac{\Delta t}{\rho} \Delta x$

$$\Delta_x = \frac{X_{max} - X_{min}}{N_x - 1}, \Delta t = \frac{t_{max} - t_{min}}{N_t - 1}$$

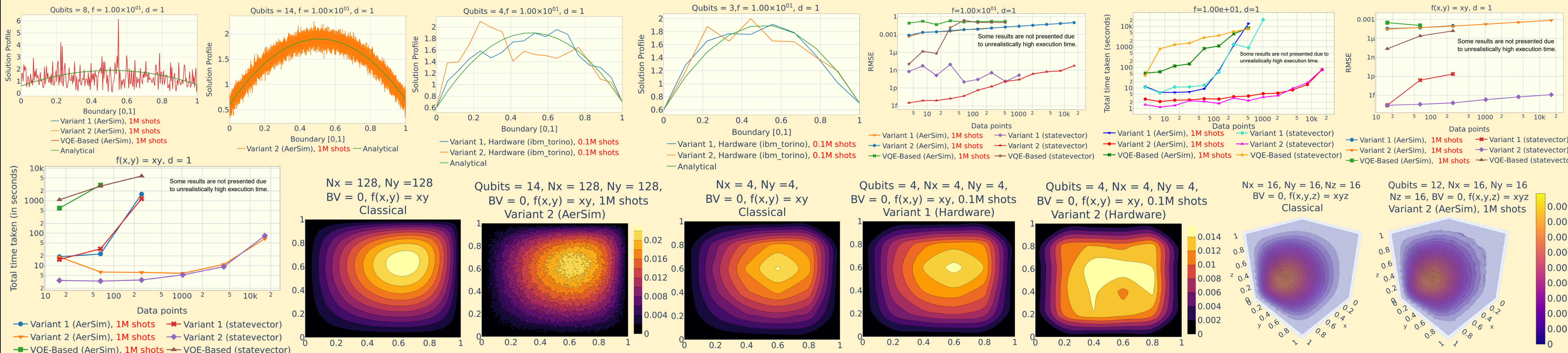


Applications: Computational fluid dynamics (CFD), aircraft and automotive design, atmospheric circulation, cardiovascular studies

Experimental Results

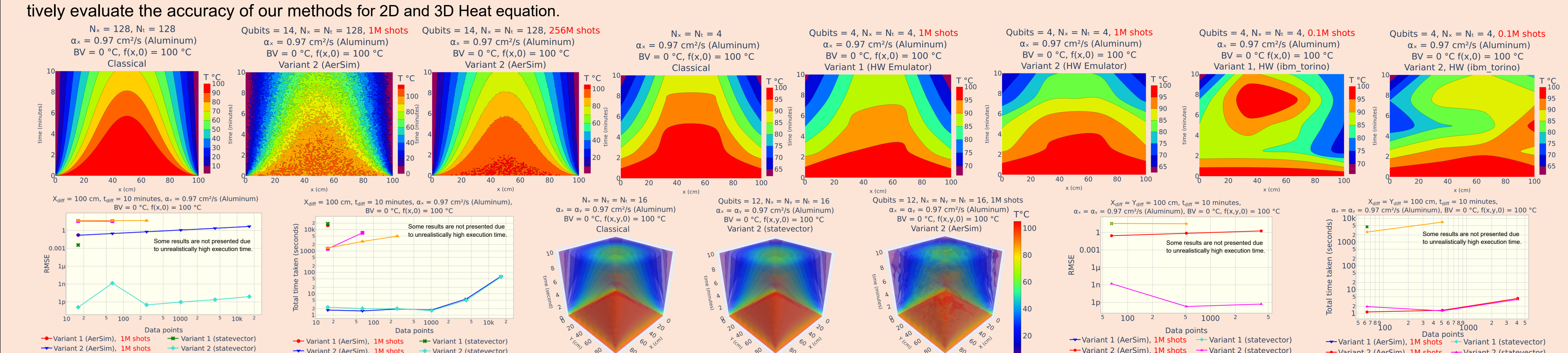
Case Study 1: Multidimensional Poisson Equation

- We have provided the solution profiles, total execution times, and accuracy results obtained using our proposed algorithm, and compared them to the VQE-based approach on noise-free and noisy simulators, as well as on real quantum hardware (ibm_torino). We have used root mean square error (RMSE) as a metric to quantitatively evaluate the accuracy of our methods for 1D, 2D, and 3D Poisson equation.



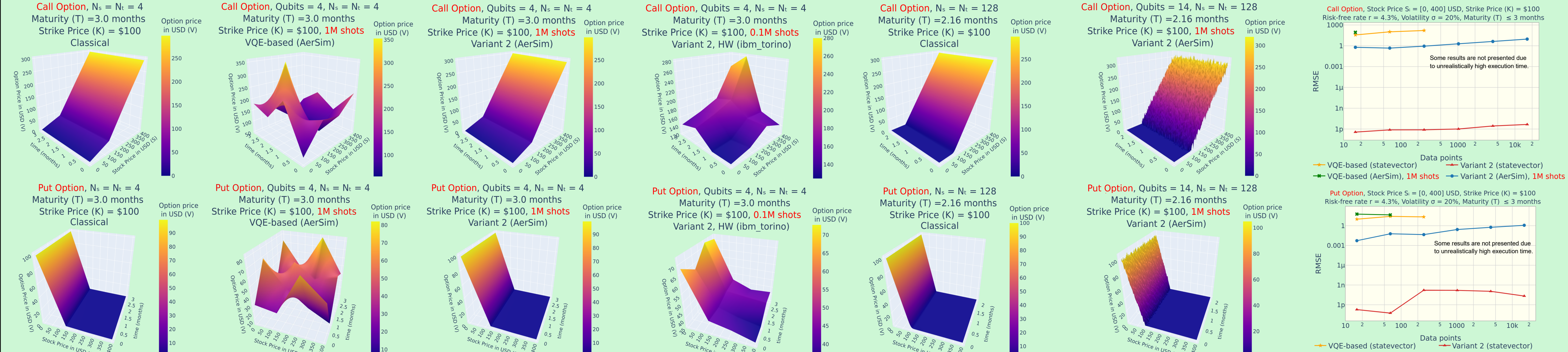
Case Study 2: Multidimensional Heat Equation

- We also evaluated the Heat equation using our proposed algorithm on noise-free and noisy simulators, as well as on hardware emulator (FakeTorino) and real quantum hardware. Experimental results demonstrate favorable performance in terms of accuracy, scalability, and execution time for 2D, 3D, and 4D space-time Heat equations. We have used root mean square error (RMSE) as a metric to quantitatively evaluate the accuracy of our methods for 2D and 3D Heat equation.



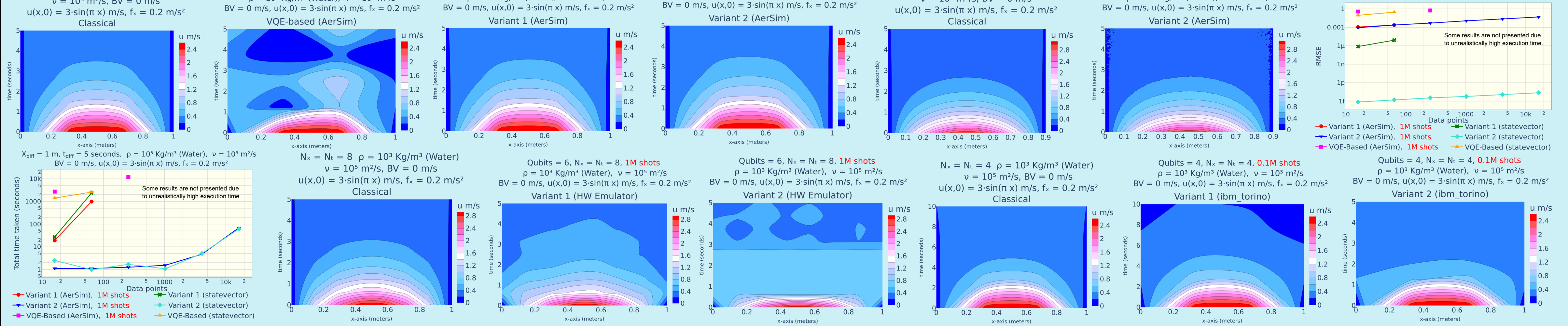
Case Study 3: Black-Scholes Equation

- We also evaluated the Black-Scholes equation for both call and put options using our proposed algorithm on noise-free and noisy simulators, as well as on hardware emulator (FakeTorino) and real quantum hardware. Experimental results demonstrate favorable performance in terms of accuracy, scalability, and execution time for both call and put options.



Case Study 4: Navier-Stokes Equation

- We also evaluated the 2D Navier-Stokes equation using our proposed algorithm on noise-free and noisy simulators, as well as on hardware emulator (FakeTorino) and real quantum hardware.



Experimental Setup

- Quantum Framework:**
 - Qiskit v1.1.0 [5]
 - BQSKIT v1.1.0 [3]
- Techniques:**
 - Proposed Algorithms
 - Variant 1 (BQSKIT)
 - Variant 2 (Qiskit)
- Metrics:**
 - Scalability quantified in terms of number of datapoints, number of data dimensions, and number of qubits
 - Accuracy quantified in terms of root-mean-square-error (RMSE)
 - Total execution time
- Simulators and Emulators:**
 - Noise-free (staveator simulator)
 - Noisy (AerSimulator)
- Hardware Emulator (FakeTorino)**
- Quantum hardware (ibm_torino)**
- Circuit samples (shots):** 1M, 256M, and 0.1M
- Hardware Specifications:**
 - Intel Xeon Gold 6342 CPU
 - 48 cores @ 2.8GHz
 - 3x NVIDIA A100 80GB GPUs
 - 256GB DDR4 RAM @ 3200MHz
- PDEs Tested:**
 - Poisson equation
 - 1 dimensional (1D)
 - $f(x) = 10, d=1, BVs = -0.6$ and -0.7
 - 2 dimensional (2D)
 - $f(x,y) = xy, d=1, BVs = 0$
 - 3 dimensional (3D)
 - $f(x,y,z) = xyz, d=1, BVs = 0$
 - Heat equation
 - 2 dimensional (2D)
 - Initial value $f(x,y,0) = 100^\circ C$, $X_{up} = 100$ cm, $BVs = 0^\circ C$, Thermal diffusivity $\alpha = 0.97 \text{ cm}^2/\text{s}$
 - 3 dimensional (3D)
 - Initial value $f(x,y,z,0) = 100^\circ C$, $X_{up} = Y_{up} = Z_{up} = 100$ cm, $BVs = 0^\circ C$, thermal diffusivity $\alpha_x = \alpha_y = \alpha_z = 0.97 \text{ cm}^2/\text{s}$
 - Black-Scholes equation
 - Call and put option
 - Risk free interest rate $r = 4.3\%$, strike price $K = \$100$, volatility $\sigma = 20\%$, $BVs = 0^\circ C$, initial stock price range $[0, 400]$ [11]
 - Navier-Stokes equation
 - Kinematic viscosity $\nu = 10^{-6} \text{ m}^2/\text{s}$ for water, $X_{up} = 1$ m, density $\rho = 1000 \text{ Kg/m}^3$, external force $F = 0.2 \text{ m/s}^2$

Conclusions and Future Work

- In this work, we have proposed two variants of an efficient and scalable quantum solver algorithm for solving multidimensional partial differential equations (PDEs).
- The first variant, i.e., Variant 1, uses classical-to-quantum (C2Q) encoding, the finite difference method (FDM), and numerical instantiation. While, the second variant, i.e., Variant 2, integrates C2Q, FDM, and column-by-column decomposition (CCD).
- As a case study, we have applied our algorithm to the multidimensional Poisson, Heat, Black-Scholes, and Navier-Stokes equations.
- We have compared our results with those from VQE-based PDE solvers on noise-free and noisy simulators as well as on real quantum processors (QPU).
- The experimental results demonstrate favorable performance in terms of accuracy, scalability, and execution time compared to existing PDE solvers based on quantum variational algorithms.
- This work showcases robust potential of quantum computing for solving multidimensional PDEs.
- We anticipate that the ongoing research in quantum synthesis and data encoding could be very beneficial to demonstrating the efficiency and practicality of our algorithms for solving PDEs on real quantum hardware, particularly NISQ devices.
- In future work, we will be working on improving the scalability, execution time, and accuracy of our proposed algorithm while maintaining its applicability to multidimensional problems and ensuring practical utility.

Acknowledgements
This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725. This research also used resources of the National Energy Research Scientific Computing Center (NERSC), a Department of Energy Office of Science User Facility under Contract DE-AC02-05OR21464.

References

- El-Araby, E., Mahmud, N., Jeng, M.J., MacGillivray, A., Choudhary, M., Nobel, M.A.L., Islam, S.I.L., Levy, D., Kneidel, D., Watson, M.R. and Bauer, J.G., 2023. Towards complete and scalable emulation of quantum algorithms on high-performance reconfgurable computers. *IEEE Transactions on Computers*, 72(8), pp.2550-2564.
- Shende, Vivek V., Stephen S. Bullock, and Igor L. Markov. "Synthesis of quantum logic circuits." In Proceedings of the 2005 Asia and South Pacific Design Automation Conference, pp. 272-275. 2005.
- Younis, Ed. et al. *Berkeley quantum synthesis toolkit (bqskit) v1*. No. Berkeley Quantum Synthesis Toolkit. Lawrence Berkeley National Laboratory (LBNL), Berkeley, CA (United States), 2021.
- Iran, Babar, and Roger Colbeck. Ivan Kukuljan, Jonathan Home, and Matthias Christandl. "Quantum circuits for isome-theory." *Physical Review A*, 95, 3 (2016): 032318.
- IBM Quantum. Qiskit: An Open-Source Framework for Quantum Computing. Available online: https://www.ibm.com/quantum/qiskit (accessed on 28 February 2025).
- Ali, Mazen, and Matthias Kabel. "Performance study of variational quantum algorithms for solving the Poisson equation on a quantum computer." *Physica Review Applied* 20.1 (2023): 014054.
- Sato, Y., Kondo, R., Koide, S., Takamitsu, H., & Imoto, M. (2021). Variational quantum algorithm based on the minimum potential energy for solving the Poisson equation. *Physical Review A*, 104(6), 063409.
- Childs, Andrew M., Robin Kothari, and Roberto D. Somma. "Quantum algorithms for systems of linear equations with exponentially improved dependence on precision." *SIAM Journal on Computing* 46 (2017): 1920-1950.
- Pollacco, G., Salazar, J. P., G6es, G. B., Maciel, T. O., & Duzovnik, E. I. hybrid classical-quantum approach to solve the heat equation using quantum annealers. *Phys. Rev. A*, 104, 032426 (2021).
- Tavella, D., & Randall, C. Pricing Financial Instruments: The Finite Difference Method. Wiley (2000).