Quantum-Classical Hybrid Algorithms for Darcy Flow with Local Mass Conservation *

Jiangguo Liu[1]

Department of Mathematics, Colorado State University, Fort Collins, CO 80523-1874, USA, jiangguo.liu@colostate.edu

Abstract. This paper investigates quantum-classical hybrid algorithms for 1-dim Poisson equation and/or Darcy flow. We consider mainly two numerical methods for spatial discretizations: the central finite difference method (CtrFD) and a novel 1-dim weak Galerkin finite element method (WGFEM). The latter is preferred due to its local mass conservation property. The methods are implemented in `Python` utilizing `Qiskit 2.3.0` libraries. Numerical experiments are conducted with examination focused on fidelity, reduction of the cost function, and residual evolution.

Keywords: Darcy flow · Fidelity · Local mass conservation · Quantum-classical hybrid algorithms · Weak Galerkin Finite Element Methods

1 Introduction

The Poisson equation, along with somehow the more general Darcy equation, are basic types and also relatively easier partial differential equations for development of algorithms for quantum computing. Although research efforts on quantum algorithms for such equations in multi-dimensional spaces can be observed in the literature, 1-dim problems still need to be studied for better (more robust) algorithms that work well on the Noisy Intermediate-Scale Quantum (NISQ) or near-term computers.

The algorithm developed in [17] applies to heterogenous Darcy flow in 3D (with fractures). Block encoding was applied to the stiffness matrix obtained from finite difference discretization. Complexity $\mathcal{O}(N^{2/3}polylog(N)\log\epsilon^{-1})$ was attained, outperforming the best classical solver $\mathcal{O}(N\log N\log\epsilon^{-1})$. Effective preconditioning was pointed out as a key barrier in achieving quantum advantages. But the finite difference method is not mass-conservative.

The work in [4] is about quantum realization of the Lagrangian $\mathcal{Q}_1$ finite elements on d-dim Cartesian grids for the Poisson equation via block encoding and the quantum singular value transformation (QSVT) (for pseudoinverse). The optimal runtime $\mathcal{O}(\sqrt{\kappa}\,\epsilon^{-1}polylog(\kappa\epsilon^{-1}))$ (κ as the condition number) is proved and a BPX preconditioner is investigated. Numerical results based on `Qiskit` implementation on a noiseless simulator are presented, although no implementation

* J. Liu was partially supported by US National Science Foundation under grant DMS-2208590.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_13

detail is available. Dependence on realistic noise is studied, but the algorithm is not really designed for NISQ computers.

The **Qu-FEM** (Lagrangian elements on Cartesian grids) in [2] is also for fault-tolerant computers only.

As is well known, numerical PDE solvers rely on linear solvers. As for quantum linear solvers, we refere to [5,16]. For practical implementation of HHL in `Python`, we refer to

- `https://github.com/LeDernier/qiskit-HHL?tab=readme-ov-file`
  This implementation by LeDernier was based on [5], utilizing `Qiskit 0.34.0, Qiskit-Aqua 0.9.5, NumPy 1.21.5`, and of course `Python 3`.

- `https://github.com/Qiskit/textbook/blob/main/notebooks`
  It contains a nice tutorial, but code scalability is unknown.

Needless to say, many existing papers are related to what we intend to study. [3,9,19] are about quantum Poisson solvers. [5,6,16] are about quantum linear solvers. Quantum finite element methods can be found in [4,2], while [17] deals with 3D/real problems. [1] intended to establish a commercial package. A set of comprehensive lecture notes on quantum computing was presented [8]. `QuantikZ`, a nice utility software related to quantum computing, was introduced in [7].

Efficient numerical solvers with physical-property-preserving for Darcy flow and related problems have been our research focus [10,11,12,13,14,18,20,21,22]. The success of our numerical methods and more realistic and challenging 3-dim problems motivate us to study quantum algorithms for these property-preserving numerical methods, especial that work well on the NISQ computers.

The rest of this paper is organized as follows. Section 2 briefly presents basic concepts of quantum computing. Section 3 recalls the finite difference method, then elaborate on a novel weak Galerkin finite element method for 1-dim Darcy flow and outline a hybrid algorithm for this numerical method. Section 4 discusses variational quantum linear solvers (VQLS). Section 5 discusses implementation of the algorithm using `IBM Qiskit` library. Section 6 presents numerical results. Section 7 concludes the paper with remarks for future work.

2 Basic Concepts of Quantum Computing

Basic concepts of quantum computing

- *Qubit*: The basic unit of information on a quantum computer is the quantum bit (*qubit*). The state of an m-qubit quantum computer is a unit vector in the linear space $\mathbb{C}^{2^m}$, which has 2^m standard unit vectors. They are written as, by adopting the Dirac bra-ket notations,

$$|0\cdots00\rangle, \quad |0\cdots01\rangle, \quad \cdots, \quad |1\cdots11\rangle.$$

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_13

For instance, a 1-qubit state is expressed as

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle = \alpha_0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \alpha_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \text{ where } \alpha_0, \alpha_1 \in \mathbb{C}, \quad |\alpha_0|^2 + |\alpha_1|^2 = 1.$$

- Corresponding to m qubits, there are 2^m standard basis vectors and then all possible superpositions of these standard states. For example, with 34 qubits, one can handle a linear system with size $2^{30} = 1,073,741,824$ (about 1 billion). So quantum computers are expected to be much more powerful than classical computers.

- *Operations* evolving the state are realized through *quantum circuits*, which correspond to unitary matrices $U \in \mathbb{C}^{2^m \times 2^m}$. Quantum circuits consists of *quantum gates*, e.g., some 2×2 unitary matrices. Composition of basic gates correspond to matrix multiplication and tensor products.

- A *measurement* of the state $|\psi\rangle = \sum_{j=0}^{2^m-1} \alpha_j |j\rangle$ is not a unitary operation. The outcome is a random variable with sample space $\{0,1\}^m$ and the probability for $X = j$ is just $|\alpha_j|^2$.

Noisy intermediate-scale quantum (NISQ) versus **Fault-tolerant quantum computers**. Ideally, we would have enough qubits for storing information, the quantum circuits have enough depth, and the noise is under control. Currently, `Regatti Novera` has 9 qubits and the system is available for the scientific research community. `IBM Heron` has 156 qubits, `IBM Condor` has 1,121 qubits, and `IBM` plans to have a 1 million-qubit quantum computer installed by 2030. Some of `IBM` and `Regatti` quantum hardwares are available with costs via cloud services provided by `BlueQubit`.

3 FDM/FEM for 1-dim Darcy Flow

3.1 The Finite Difference Method and Equivalent CG.P_1

Assume the domain is the standard unit interval $[0,1]$. The 1-dim Poisson with homogeneous BVP is posed as $-p''(x) = f(x), p(0) = p(1) = 0$.

Consider a uniform partition with $h = \frac{1}{n}$, $x_i = ih$, $i = 0, 1, \cdots, n$. There are totally $(n+1)$ nodes, $n-1$ interior nodes, or n elements (small intervals). Two numerical methods can be applied.

- The central finite difference (CtrFD);
- The continuous/conforming Galerkin FEM with Lagrangian P_1 elements.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_13

It is well known that the resulting linear systems are the same, as shown below.

$$\begin{bmatrix} 2 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & 1 & & & \\ & & \cdots & \cdots & \cdots & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 2 & -1 \\ & & & & & -1 & 2 \end{bmatrix} \begin{bmatrix} p_{h,1} \\ p_{h,2} \\ \cdots \\ p_{h,i} \\ \cdots \\ p_{h,n-2} \\ p_{h,n-1} \end{bmatrix} = h^2 \begin{bmatrix} f_1 \\ f_2 \\ \cdots \\ f_i \\ \cdots \\ f_{n-2} \\ f_{n-1} \end{bmatrix}. \tag{1}$$

This size $(n-1)$ linear system is also written generally as $\mathbf{Ax} = \mathbf{b}$ and will be solved by HHL or a variational quantum linear solver.

A general Dirichlet boundary value problem for 1-dim Darcy flow posed as

$$-(K(x)p')' = f(x), \;\; x \in (a,b), \qquad p(a) = p_a, \;\; p(b) = p_b, \tag{2}$$

for which a partition with n elements ($(n-1)$ interior nodes, $(n+1)$ all nodes) $a = x_0 < x_1 < \cdots x_{i-1} < x_i < x_{i+1} < \cdots < x_n = b$ is not necessarily uniform. Denote $h_i = x_i - x_{i-1}$ for $i = 1, \cdots, n$. Note that we have N elements and deal with $(n-1)$ unknowns $p_h^{(i)}$ (also written as $p_{h,i}$) for $i = 1, \cdots, n-1$.

For the finite difference method, we have approximation at the interior nodes

$$\frac{\Big(-K_{i+1}\dfrac{p_h^{(i+1)} - p_h^{(i)}}{h_{i+1}}\Big) - \Big(-K_i \dfrac{p_h^{(i)} - p_h^{(i-1)}}{h_i}\Big)}{\frac{1}{2}(h_i + h_{i+1})} = f_i, \quad i = 1, \ldots, n-1. \tag{3}$$

which can be rewritten as, for $1 \le i \le n-1$,

$$-\frac{2h_{i+1}K_i}{h_i + h_{i+1}} p_h^{(i-1)} + 2\frac{h_{i+1}K_i + h_i K_{i+1}}{h_i + h_{i+1}} p_h^{(i)} - \frac{2h_i K_{i+1}}{h_i + h_{i+1}} p_h^{(i+1)} = h_i h_{i+1} f_i. \tag{4}$$

Here K_i, K_{i+1} are understood as the elementwise averages of the conductivity on $[x_{i-1}, x_i]$ and $[x_i, x_{i+1}]$. We end up with a tridiagonal linear system, which is generally ***unsymmetric*** unless we adopt a uniform mesh.

It is interesting to see that for the conforming Galerkin finite element method with Lagrangian P_1 elements, we obtain ***the same equations*** in (4).

The velocity on each element $E_i = [x_{i-1}, x_i]$ is actually a constant:

$$u|_{E_i} = -K_i(p_h^{(i)} - p_h^{(i-1)})/h_i, \quad i = 1, \cdots, n. \tag{5}$$

The flux discrepancy at the interior node x_i is

$$\Delta(\mathbf{u} \cdot \mathbf{n})|_{x_i} := u|_{E_i} - u|_{E_{i+1}}, \quad 1 \le i \le n-1. \tag{6}$$

The fluxes at the boundary nodes are computed as,

$$\texttt{FluxLeftBndry} = K_1 \frac{p_h^{(1)} - p_a}{h_1}, \quad \texttt{FluxRightBndry} = -K_n \frac{p_b - p_h^{(n-1)}}{h_n}. \tag{7}$$

The flux discrepancy at the interior nodes form a size $(n-1)$ vector with maximal value at $\mathcal{O}(h)$. Its norm can be set as an ***observable*** for quantum computing.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_13

3.2 Weak Galerkin FEM WG($P_0, P_0; P_1$) for 1-dim Darcy Flow

Now we consider 1-dim Darcy flow with Dirichlet boundary conditions

$$\begin{cases} \partial_x(-K(x)\,\partial_x p) =: \nabla \cdot u = f(x), \quad x \in \Omega = (a,b) \\ p(a) = p_a, \;\; p(b) = p_b \end{cases} \tag{8}$$

For local (and hence global) mass conservation, we develop a 1-dim weak Galerkin (WG) finite element method to discretization the Darcy equation. Our WG FEM is different and more intuitive than that in [23].

Let $\mathcal{E}_h : a = x_0 < x_1 < \cdots < x_{i-1} < x_i < x_{i+1} < \cdots < x_n = b$ be a (not necessary uniform) partition of $\overline{\Omega} = [a,b]$. Clearly, we have n elements and $n+1$ nodes. We put constant unknowns at all nodes and also on all elements, so we have totally $2n+1$ pressure unknowns.

For a typical element $E = [x_{i-1}, x_i]$, $1 \le i \le n$, we set $x_c = \frac{x_{i-1}+x_i}{2}$ and $X = x - x_c$ (as the centralized coordinates). Denote $h = x_i - x_{i-1}$. Consider $P_1(E) = \texttt{Span}(1, X)$. Its Gram matrix is a diagonal matrix $\texttt{diag}(h, \frac{h^3}{12})$. Then we consider three weak Galerkin local basis functions:
– ϕ_0: taking value 1 in element interior but 0 at both endpoints;
– ϕ_l: taking value 0 in element interior, 1 at left endpoint, 0 at right endpoint;
– ϕ_r: taking value 0 in element interior, 0 at left endpoint, 1 at right endpoint.
Their discrete weak gradients are constructed in the above $P_1(E)$ space:

$$\nabla_w \phi_\circ = 0 + \frac{-12}{h^2} X, \qquad \nabla_w \phi_l = \frac{-1}{h} + \frac{6}{h^2} X, \qquad \nabla_w \phi_r = \frac{1}{h} + \frac{6}{h^2} X, \tag{9}$$

which manifests the *partition of unity*. Accordingly, the element stiffness matrix is a 3×3 symmetric and singular:

$$S_{E_i} = \left[\int_{x_{i-1}}^{x_i} K_i \nabla_w \phi \nabla_w \psi \right] = \frac{2K_i}{h_i} \begin{bmatrix} 6 & -3 & -3 \\ -3 & 2 & 1 \\ -3 & 1 & 2 \end{bmatrix}. \tag{10}$$

Such small matrices are assembled into a global stiffness matrix of order $2n+1$ that is still singular. But enforcement of the Dirichlet boundary conditions yields a ***pentadiagonal*** nonsingular matrix $\mathbf{S}_{\mathcal{E}_h}$ with a condition number $\mathcal{O}(h^{-2})$. This is different than the tridiagonal matrix generated by the standard central finite difference method. Shown in Fig.1 is the sparsity structure of such a matrix when $n = 16$.

After enforcing the Dirichlet boundary conditions, the above global linear system is reduced to size $2n-1$. It can be solved by an appropriate direct or iterative or variational linear solver to obtain a numerical pressure vector p_h (size $2n-1$) that consists of two groups: $\mathtt{p}_h^{(i)}, i = 1, \cdots, n$ for the elements, and $p_h^{(j)}, j = 1, \cdots, n-1$ for the interior nodes.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_13

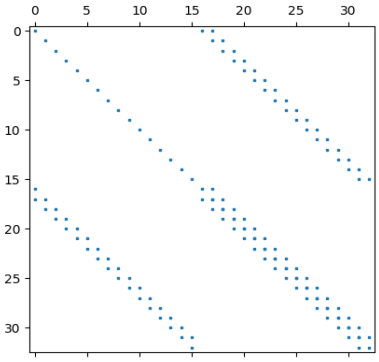

Fig. 1. An illustration of the 5-diagonal sparsity structure of the global stiffness matrix of WG$(P_0, P_0; P_1)$ for 1-dim Darcy equation with $n = 16$ elements.

For a typical element $E_i = [x_{i-1}, x_i], 1 \le i \le n$, we retrieve the numerical pressure ***interior value*** $\mathtt{p}_h^{(i)}$ and the nodal values as $p_h^{(i-1)}, p_h^{(i)}$. Based on the discrete weak gradient $\nabla_w p_h$, the numerical Darcy velocity v_h is

$$v_h|_{E_i} = -K_i \left(p_h^{(i-1)} \frac{-1}{h} + p_h^{(i)} \frac{1}{h} \right) - K_i \left(\mathtt{p}_h^{(i)} \frac{-12}{h^2} + p_h^{(i-1)} \frac{6}{h^2} + p_h^{(i)} \frac{6}{h^2} \right) X. \quad (11)$$

The numerical velocity at the midpoint (in agreement with our intuition) is

$$v_h|_{x_c} = -K_i \frac{p_h^{(i)} - p_h^{(i-1)}}{h}, \quad i = 1, \cdots, n. \quad (12)$$

The numerical fluxes at the left endpoint x_{i-1} and the right endpoint x_i are, respectively,

$$\begin{aligned} F^{(i)}|_{x_{i-1}} &= K_i \frac{p_h^{(i)} - p_h^{(i-1)}}{h_i} - 3K_i \frac{p_h^{(i)} - 2\mathtt{p}_h^{(i)} + p_h^{(i+1)}}{h_i}, \\ F^{(i)}|_{x_i} &= -K_i \frac{p_h^{(i)} - p_h^{(i-1)}}{h_i} + 3K_i \frac{p_h^{(i)} - 2\mathtt{p}_h^{(i)} + p_h^{(i+1)}}{h_i}. \end{aligned} \quad (13)$$

The flux discrepancy at an interior node $x_i, i = 1, \cdots, n-1$ is theoretically zero but computed as the algebraic sum of the nodal fluxes. The numerical fluxes at the left and right endpoints are

$$\begin{aligned} \mathtt{Flux}|_{x=a} &= K_1 \frac{p_h^{(1)} - p_a}{h_1} + 3h_1 K_1 \frac{p_a - 2\mathtt{p}_h^{(1)} + p_h^{(1)}}{h_1^2}, \\ \mathtt{Flux}|_{x=b} &= -K_n \frac{p_b - p_h^{(n-1)}}{h_n} - 3h_n K_n \frac{p_h^{(n-1)} - 2\mathtt{p}_h^{(n)} + p_b}{h_n^2}, \end{aligned} \quad (14)$$

where the 2nd terms reflect the correction utilizing the central differences. The flux discrepancy at the interior nodes (***theoretically zero***) form a size $(n-1)$ vector also. Its norm can also be set as an ***observable*** for quantum computing.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_13

We point that Schur complement and soft enforcement of Dirichlet boundary conditions [11] can be incorporated into our implementation. As such, the global stiffness matrix will be the same as that for the central finite difference method when $K(x) = 1$ and a uniform grid is used.

3.3 A Hybrid Algorithm for WG Solver for 1d Darcy Flow

Anyway, we end up with a large-scale sparse linear system

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{15}$$

Instead of applying the HHL algorithm, which requires a deep quantum circuit, we consider the variational approach that can be applied to NISQ computers.

4 Variational Quantum Linear Solvers (VQLS)

For either FD/CG.P_1 or WG$(P_0, P_0; P_1)$ applied to 1-dim Darcy flow, we obtain a sparse linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$, where the coefficient matrix $\mathbf{A}$ is real symmetric and hence Hermitian. The linear system can be be solved by HHL or a variational quantum linear solver (VQLS).

For the variational/hybrid approach, a classical minimizer is employed to minimize a cost function and adjust the parameters of a quantum circuit. In general, there are two categories of optimiers: gradient-free and gradient-based methods.

Gradient-free optimizers

- *Constrained Optimization By Linear Approximation* (COBYLA):
  Fast convergence in noise-free simulations, relatively well with some noise;
- *Simultaneous Perturbation Stochastic Approximation* (SPSA):
  Widely used, most robust and best-performing optimizer in noisy conditions.

Gradient-based optimizers

- *Conjugate Gradient* (CG):
  Effective in ideal scenarios but may be sensitive to noise;
- *Broyden-Fletcher-Goldfarb-Shanno* (BFGS):
  A 2nd-order method, best-performing in ideal or low-noise scenarios.

`Qiskit` offers several classical optimizers, including wrappers for `scipy.optimize`, e.g., `COBYLA`, and specialized ones, e.g., `SPSA`, used to train quantum algorithms like VQE.

For our computational tasks in this paper, we consider the following the cost function (not requiring the coefficient matrix to be symmetric/Hermitian)

$$\mathcal{C}(\psi) = 1 - \frac{|\langle b|A|\psi\rangle|^2}{\langle\psi|A^T A|\psi\rangle}, \tag{16}$$

where b is the normalized state vector for the RHS.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_13

5 `Python` Implementation Using `Qiskit` Library

The algorithms developed in this paper have been implemented in `Python` utilizing functions in the well known `IBM Qiskit 2.3.0` library.

Note that a VQLS is a actually quantum-classical hybrid algorithm, it relies on the **Estimator** primitive to compute the cost function and the **Circuit** library to construct a variational ansatz. In particular,

- `Qiskit` **SDK**: The core package required for circuit construction, pulse scheduling, and accessing quantum primitives.
- `qiskit.primitives`: Specifically, `Estimator` or `EstimatorV2` can be used to compute the expectation values required for the VQLS cost function.
- `qiskit.circuit.library`: For selecting an ansatz, e.g., `EfficientSU2` or `RealAmplitudes` that handles the trial solution for the linear system.
- `qiskit.quantum_info`: For decomposing the coefficient matrix **A** into a weighted sum of Pauli operators, as a prerequisite for the VQLS algorithm.
- `qiskit_algorithms.optimizers`: This library, although no longer bundled in the core SDK of `qiskit` 2.x, provides the classical optimizers, e.g., COBYLA, SPSA, needed to update the variational parameters.

Shown in Figure 2 is an illustration of the hybrid approach.

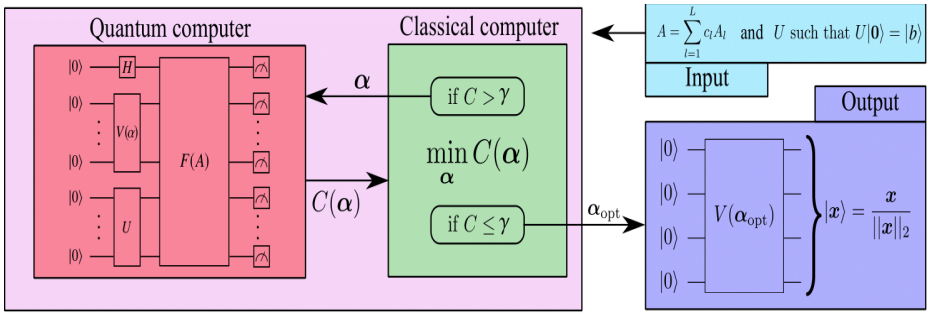

Fig. 2. Illustration of an quantum-classical hybrid algorithm.

As for hardware and software resources available for quantum computing,

- `IBM` Quantum Heron (133 qubits) is the state-of-the-art for utility-scale workloads, offering significantly lower error rates than the older generations;
- `IBM` Quantum Eagle (127 qubits) remains a workhorse of the fleet, with multiple systems like ibm_kyiv and ibm_brisbane widely available for research;
- `IBM` Quantum System Two is the hardware platform housing these latest processors, designed for modular scaling.

Regarding the access tiers: the aforementioned systems are divided into
– "Open" (free access, typically 5-27 qubits) and
– "Premium" (utility-scale, 127+ qubits).

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_13

There are **two types of accesses**.

- `IBM` Quantum Platform: The primary web interface for managing jobs and viewing system status (calibration data, error rates);
- `IBM` Cloud (Qiskit Runtime): A pay-as-you-go model that allows users to rent time on premium systems without a long-term contract.

`IBM` Quantum Computing Resources webpage shows the live status and error rates of specific backends. Such info is helpful before submitting jobs.

- Use `primitives` in `qiskit_ibm_runtime.SamplerV2` or `EstimatorV2` ...;
- Heron-based systems, e.g., ibm_torino, usually provides high-fidelity gates and hence better accuracy.
- Use faster-access 5/27 qubit systems for debugging and then the 127-qubit systems for simulations.

Note that as for `Qiskit Runtime`,

- All cloud-accessible systems are now optimized to run via `Qiskit Runtime`, which uses a primitives-based execution model (`Sampler` and `Estimator`).
- Smaller 5-qubit and 27-qubit systems (Falcon and Canary processors) are typically available on the Open Plan for educational purposes.

6 Numerical Experiments

Quantum computing (QC) is quite different than classical computing, for which accuracy of numerical methods measured in the L^2-, H^1-norms or alike can be examined through tests on popular benchmarks with known exact solutions. Efficacy further consider the computable costs needed to reach certain accuracy. For QC, however, we don't expect to extract a full numerical solution on a possibly fine mesh. Instead, we are concerned with *circuit depth*, *noise*, *observables*, *fidelity*. As PDEs are discretized into large-scale linear systems, the efficiency and/or convergence of HHL algorithms and variational quantum linear solvers (VQLS), e.g., COLABY, are further examined.

- *Fidelity*;
- *Reduction in the cost function*.

These guidelines are followed in our numerical experiments on 1-dim Poisson equation (Darcy flow).

For 1-dim Poisson equation, popular discretization methods include the continuous Galerkin with Lagrangian linear elements CG.P1, which is actually equivalent to the central finite difference method (even with a nonuniform grid). The numerical method is not locally mass-conservative, though. Either way, the discrete linear system is tridiagonal, and symmetric when a uniform mesh is used. Our new weak Galerkin finite elements WG($P_0, P_0; P_1$) is *locally mass conservative* by design. For 1-dim problem, we end up with a 5-diagonal matrix.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_13

For our novel WG($P_0, P_0; P_1$) finite element method, `Python` COO sparsity structure is adopted for assembly of the global stiffness matrix, then the matrix is converted to the CSR sparsity structure, which is more efficient for matrix-vector multiplication used in optimization algorithms, e.g., `COBYLA`.

Quantum computing can be executed on real quantum computers or simulated on local classical computers. In either scenario, "shots" refer to the number of times an algorithm is executed on a quantum circuit to sample measurement outcomes, since quantum measurements are probabilistic. Each run collapses the quantum state, giving us a classical bit string sampled from the probability distribution defined by the circuit's final state.

On real quantum hardware, e.g., IBM Quantum, one needs many shots (1,000?10,000+, by the Law of Large Numbers) to get reliable statistics and estimate expectation values accurately to overcome inherent hardware errors and (shot) noise.

However, when running simulations on a local machine (CPU/GPU) utilizing, e.g., using Qiskit libraries, one do not need many shots. This is because classical computers can exactly track the quantum state for small-to-medium circuits (up to 30 or 40 qubits).

In principle, local simulations serve as a powerful tool for prototyping quantum algorithms priori to cloud/hardware runs.

Table 1. Ex.1 Test II: VQLS w/ COLYBA, #MaxItr=1200, Threshold=10^{-12}

#Interior grid points	#Qubits	Condition number	#Iterations	Fidelity
4	2	9.472×10^0	988	1.0000
8	3	3.216×10^1	1009	0.9687
16	4	1.164×10^2	1017	0.7934
32	5	4.406×10^2	912	0.6263
64	6	1.711×10^3	892	0.6993
128	7	6.743×10^4	878	0.6499

This section presents numerical and graphical results on one well-known example.

Example 1. This popular/standard example deals with the 1-dim Poisson equation on the unit interval $\Omega = (0, 1)$ with the homogeneous Dirichlet boundary conditions. In particular, $K(x) = 1$, $f(x) = \pi^2 \sin(\pi x)$, and the exact solution is well known as $p(x) = \sin(\pi x)$. Actually, we have performed two tests.

– Test I: The discrete linear system was solved by the HHL algorithm. Unfortunately, this was not successful. Since the HHL solver was removed from the the `Qiskit` library, we have to search for legacy code, some worked, some did not work. For those working code, results were unsatisfactory, especially when a grid has 8 or more interior points. `LuGo` [15] is known as

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_13

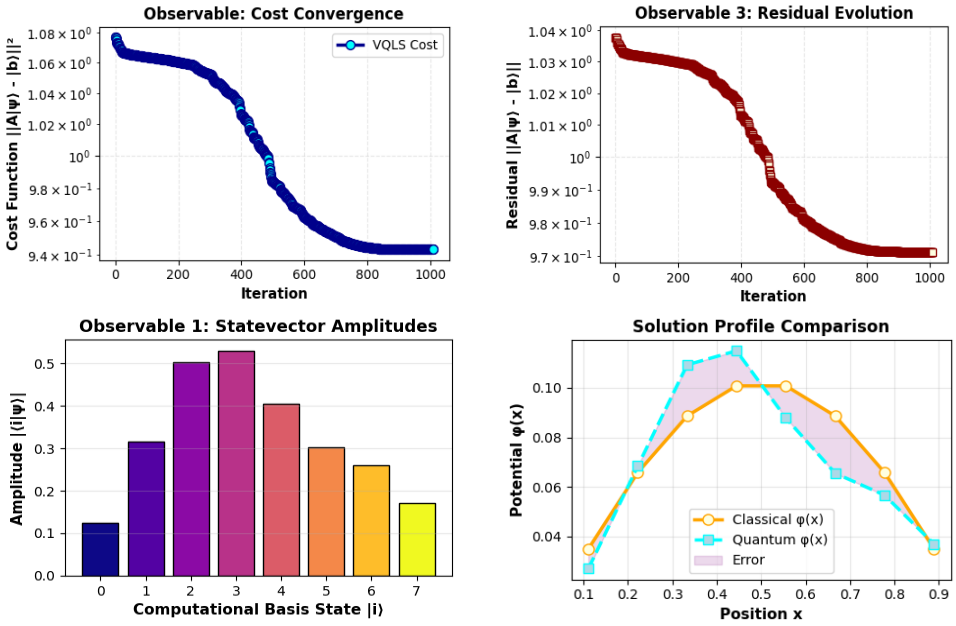

Fig. 3. Ex.1 Test II: Results of a VQLS w/ COLYBA for the WG$(P_0, P_0; P_1)$ finite element method with 8 interior grid points.

an enhanced Quantum Phase Estimation (QPE) approach that significantly optimizes the HHL algorithm for solving linear systems. But we have not been able to implement this relatively new solver yet.
– Test II: The linear system is solved by `COLYBA`, a variational quantum linear solver. This approach is similar to that discussed in [9]. The `Qiskit` library was utilized for `Python` coding.

Note also that

– `scipy.sparse.linalg.eigsh` command is used to estimate the largest and smallest eigenvalues of the stiffness matrix and then the ratios for the condition number.
– The matrix is not too large and can be converted to a dense matrix. Then `numpy.linalg.cond` is called to compute the conditional number also.

The results presented in Table 1 and Figure 3 demonstrate promise. For a small to medium mesh, the fidelity keeps above a certain level, while the cost convergence and residual evolution can be clearly observed.

7 Concluding Remarks

This paper presents our preliminary work on development of property-preserving quantum solvers for Darcy flow. The weak Galerkin finite element methods

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_13

were used due to the local mass conservation property, the symmetric-positive-definiteness of the global stiffness matrix, the use of Schur complement, and soft enforcement of Dirichlet boundary conditions. We concentrate on quantum-classical algorithms that can be executed on NISQ computers. Mathematically, we focus on the 1-dim Darcy equation, although our new locally mass-conservative weak Galerkin finite element methods can be extended to multi-dimensional Darcy flow problems. The challenges are not in the numerical methods but more on the implementation of the methods on NISQ hardwares.

The results in this paper show both promises and challenges. Further numerical experiments, e.g., noise with real hardwares, need to be done. Our research will be extended in three directions.

(i) Darcy flow problems in 2-dim or even 3-dim which exhibits stronger heterogeneity and anisotropy, and hence quantum computing could help address the challenges.
(ii) Development of hybrid-classical algorithms for transport problems that preserve physical properties, e.g., positivity of concentration of the solute being transported.
(iii) Coupling of quantum-classical hybrid algorithms for flow solvers and transport solvers will be even more challenging. How to handle the physical quantities, e.g., velocity, local fluxes, from flow solvers as observables and to pass them to transport solver are even more challenging.

Progresses in these research directions will be reported in our future work.

References

1. et al., H.J.: H-des: a quantum-classical hybrid differential equation solver. arXiv (2025)
2. Alkadri, A.M., Kharazi, T.D., Whaley, K.B., Mandadapu, K.K.: A quantum algorithm for the finite element method. arXiv **2411**, 02522v3 (2025)
3. Cao, Y., Papageorgiou, A., Petras, I., Traub, J., Kais, S.: Quantum algorithm and circuit design solving the Poisson equation. New J. Phys. **15**, 013021 (2013)
4. Deiml, M., Peterseim, D.: Quantum realization of the finite element method. Math. Comput. **In press** (2025)
5. Dervovic, D., Herbster, M., Mountney, P., Severini, S., Usher, N., Wossnig, L.: Quantum linear systems algorithms: a primer. arXiv (1802), 08227 (2018)
6. Hosaka, A., Yanagisawa, K., Koshikawa, S., Kudo, I., Alifu, X., Yoshida, T.: Preconditioning for a variational quantum linear solver **2311**, 15657v3 (2024)
7. Kay, A.: Tutorial on the Quantikz package. arXiv **1809**, 03842v7 (2023)
8. Lin, L.: Lecture notes on quantum algorithms for scientic computation. U.C. Berkeley (2022)
9. Liu, H.L., Wu, Y.S., Wan, L.C., Pan, S.J., Qin, S.J., Gao, F., Wen, Q.Y.: Variational quantum algorithm for the Poisson equation. Phys. Rev. A **104**, 022418 (2021)
10. Liu, J., Gu, J., Li, H., Carlson, K.H.: Machine learning and transport simulations for groundwater anomaly detection. J. Comput. Appl. Math. **380**, 112982 (2020)

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_13

11. Liu, J., Sadre-Marandi, F., Wang, Z.: DarcyLite: A Matlab toolbox for Darcy flow computation. Proc. Comput. Sci. **80**, 1301–1312 (2016)
12. Liu, J., Tavener, S., Wang, Z.: Lowest-order weak Galerkin finite element method for Darcy flow on convex polygonal meshes. SIAM J. Sci. Comput. **40**, B1229–B1252 (2018)
13. Liu, J., Tavener, S., Wang, Z.: Penalty-free any-order weak Galerkin FEMs for elliptic problems on quadrilateral meshes. J. Sci. Comput. **83**, 47 (2020)
14. Liu, J., Yu, B., Li, Y.: Darcylite modules for a property-preserving transport solver. Lec. Notes Comput. Sci. **15911**, 337–352 (2025)
15. Lu, C., Meena, M.G., Gottiparthi, K.C.: LuGo: an enhanced quantum phase estimation implementation. arXiv **2503**(15439) (2025)
16. Morale, M., Pira, L., Schleich, P., Koor, K., Costa, P., An, D., Aspuru-Guzik, A., Lin, L., Rebentrost, P., Berry, D.W.: Quantum linear system solvers: A survey of algorithms and applications. arXiv **2411**, 02522 (2024)
17. Pechan, A., Golden, J., O'Malley, D.: Block encoding the 3D heterogeneous poisson equation with application to fracture flow. arXiv p. 2508.07125v1 (2025)
18. Sun, S., Liu, J.: A locally conservative finite element method based on piecewise constant enrichment of the continuous galerkin method. SIAM J. Sci. Comput. **31**, 2528–2548 (2009)
19. Wang, H.M.L.Z.X., Fei, S.M.: Variational quantum algorithms for Poisson equations based on the decomposition of sparse Hamiltonians. Phys. Rev. A **108**, 032418 (2023)
20. Wang, Z., Harper, G., O'Leary, P., Liu, J., Tavener, S.: deal.II implementation of a weak Galerkin finite element solver for Darcy flow. Lec. Notes Comput. Sci. **11539**, 495–509 (2019)
21. Yu, B., Li, Y., Liu, J.: A positivity-preserving and robust fast solver for time-fractional convection-diffusion problems. J. Sci. Comput. **98**, 59 (2024)
22. Yu, B., Li, Y., Liu, J.: New solvers for coupled flow and transport on quadrilateral meshes: Property-preserving and optimal-order convergence. Comput. Math. Appl. **203**, 73–90 (2026)
23. Zhang, T., Tang, L.: A weak finite element method for elliptic problems in one space dimension. Appl. Math. Comput. **280**, 1–10 (2016)

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_13