Oral presentation | Numerical methods

Numerical methods-V

Wed. Jul 17, 2024 4:30 PM - 6:30 PM Room A

[9-A-02] Obtaining converged flow solutions using quantum annealing algorithm

*Yuichi Kuya[1], Asaga Takahito [1] (1. Tohoku University)

Keywords: Quantum annealing, Lattice gas automata (LGA) , Finite difference method

**Twelfth International Conference on
Computational Fluid Dynamics (ICCFD12),
Kobe, Japan, July 14-19, 2024**

Obtaining converged flow solutions using quantum annealing algorithm

Yuichi Kuya* and Takahito Asaga*
Corresponding author: kuya@tohoku.ac.jp * Tohoku University, Japan.

Abstract: This study discusses quantum annealing computation methods to obtain flow solutions. Quantum annealing has attracted attention in recent years as a method for finding optimal solutions to combinatorial optimization problems faster and/or more accurately than conventional methods. First, the quantum annealing lattice gas automata (qaLGA) proposed by a recent paper is introduced. We then propose numerical methods to obtain converged flow solutions by quantum annealing computation, utilizing the quantum superposition state. The proposed methods are built for lattice gas automata (LGA) and finite difference methods. In numerical experiments, both proposed methods obtain the velocity profile of a channel flow from a number of possible solutions using quantum annealing compuataion.

Keywords: Quantum Annealing, Computational Fluid Dynamics, Lattice Gas Automata, Finite Difference.

1 Introduction

In the design of fluid-mechanical systems such as aircraft, gas turbines, and hydraulic pumps, RANS (Reynolds-averaged Navier–Stokes) computations are still widely used in design because of the good balance between computation time and prediction accuracy. However, the number of studies on RANS seems to be gradually decreasing, and no innovative models that significantly improve the prediction accuracy of flow fields have emerged. For this reason, in recent years, LES (large eddy simulation), which has better prediction accuracy than RANS, and hybrid methods of RANS and LES have attracted more attention. For example, LES around a whole aircraft configuration was recently realized [1], using a non-dissipative numerical method (KEEP scheme) [2, 3] and an LES wall model [4]. However, even with LES computations using such cutting-edge numerical methods, they require not only more computation time but also far more computational resources, compared to RANS computations. In addition, it is widely recognized that the improvement in the computing performance of conventional computers using semiconductors will eventually reach a plateau. In order to bring about an innovation in computational fluid dynamics, it is important to consider a breakthrough from a completely different perspective than before.

With the growth rate of semiconductor performance predicted to slow down in the future, one of the candidates for the next generation of computers is quantum computers. Quantum computers are operated by a mechanism completely different from conventional computers and may demonstrate computing performance far exceeding conventional supercomputers [5]. The current quantum computers may be broadly categorized into gate-based quantum computers and quantum annealing computers. Since gate-based quantum computers are theoretically capable of general-purpose computations as conventional computers, they are expected to become an alternative to conventional computers. However, the number of qubits of current gate-based quantum computers is still insufficient for solving practical applications; a qubit is a two-level system in quantum computation that corresponds to a bit in conventional computation. Also, they are prone to errors during computations. Although flow computation algorithms for gate-based quantum computers were proposed by previous studies [6, 7, 8, 9, 10, 11, 12, 13], the actual performance of those algorithms remains unknown since gate-based quantum computers with sufficient stability and the number of qubits to solve practical problems do not yet exist.

On the other hand, regarding quantum annealing computers, D-Wave Systems commercialized a quantum annealing computer equipped with more than 5,000 qubits [14]. Annealing computers that simulate quantum annealing using existing digital circuit technology are also developed, i.e., simulated-annealing computers. However, unlike gate-based quantum computers, quantum annealing computers are specialized for solving combinatorial optimization problems. Therefore, although quantum annealing computers are already being used to solve practical problems, a major issue of quantum annealing computers is their limited range of applications. Regarding flow computation algorithms, to the best

of our knowledge, there exist only two prior researches that discuss quantum annealing computation methods for numerical flow simulation [15, 16].

In this paper, we first introduce the quantum annealing lattice gas automata (qaLGA), recently proposed by Kuya et al. [16]. The lattice gas automata (LGA) [17, 18, 19, 20], which is known as the predecessor of the lattice Boltzmann method, describes the flow states using only zeros and ones and thus is suitable for quantum computers. In addition, in this paper, we propose a new flow computation method that obtains converged solutions by quantum annealing computations. Convergence calculations (or time integration) are essential not only for flow computations but also for general numerical calculations. In the proposed approach, converged solutions are extracted by quantum annealing computations, utilizing the quantum superposition state, where all the flow solutions, including converged solutions, are contained. The new flow computation method is developed for LGA and finite difference methods.

The rest of this paper is structured as follows. First, in Section 2, we provide an overview of quantum annealing. In Section 3, the quantum annealing lattice gas automata (qaLGA) [16] is introduced first, and then the proposed new flow computation methods that give converged solutions by quantum annealing computations are discussed. Conclusions are given in Section 4. Note that all the numerical results shown in this paper are obtained by simulated annealing, and only two-dimensional flow fields are considered here.

2 Overview of quantum annealing

Quantum annealing computers are primarily specialized for solving combinatorial optimization problems. Specifically, they obtain the optimal solution as a combination of $\sigma_i \in \{-1, 1\}$ or $x_i \in \{0, 1\}$ that minimizes a cost function formulated in the Ising model or QUBO (quadratic unconstrained binary optimization) equation shown below:

Ising model

$$E(\{\sigma_i\}) = \sum_i h_i \sigma_i + \sum_{i<j} J_{ij} \sigma_i \sigma_j, \quad \sigma_i \in \{-1, 1\}, \tag{1}$$

QUBO

$$E(\{x_i\}) = \sum_i \sum_j Q_{ij} x_i x_j, \quad x_i \in \{0, 1\}, \tag{2}$$

where $E(\{\sigma_i\})$ and $E(\{x_i\})$ are cost functions, h_i is the local bias acting on σ_i, and J_{ij} is the coupling constants between σ_i and σ_j. Also, Q_{ij} is a matrix (QUBO matrix) indicating the strength of interaction between x_i and x_j. Ising and QUBO models become equivalent using the following relation:

$$x_i = \frac{\sigma_i + 1}{2}. \tag{3}$$

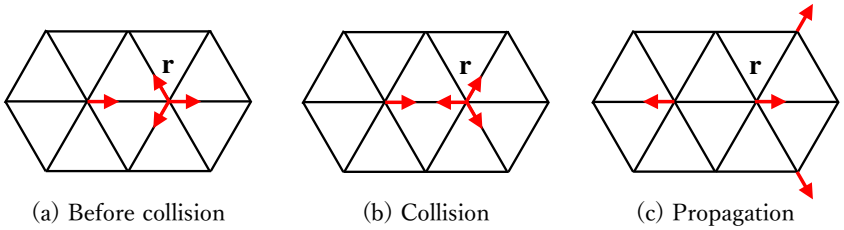
However, we develop the proposed numerical methods using the QUBO model since, as mentioned above, one of the numerical methods proposed in this study is based on LGA, where the flow states are described by zeros and ones only.

3 Flow computation by quantum annealing

3.1 Quantum annealing lattice gas automata (qaLGA) [16]

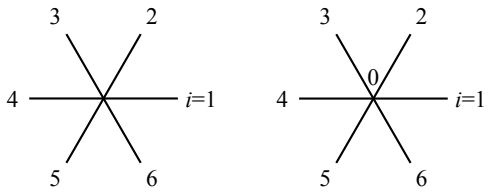
3.1.1 Overview of LGA

LGA considers virtual particles that correspond to a set of molecules or atoms and statistically processes the collision and propagation of those virtual particles to reproduce the motion of fluid flows (Fig. 1). Although previous studies proposed various numerical models and computational grid configurations, we consider the FHP (Frish-Hasslacher-Pomeau) models [17, 18, 19] here, which are widely known as the standard model of LGA. Figure 2 shows the computational grid of the FHP models. In the FHP models, a two-dimensional space is discretized into a regular hexagonal grid with a side length of 1. The



(a) Before collision (b) Collision (c) Propagation

Figure 1: LGA model: collision and propagation



(a) 6-bit model (b) 7-bit model

Figure 2: Computational grid of LGA for FHP models.

virtual particles of unit mass move between lattice nodes at unit speed along these lattice lines. The time takes integer values, and at each time, all the virtual particles exist only on the lattice nodes. The state of a lattice node at time t and position $\mathbf{r}$ is described by the following Boolean variables depending on whether or not a virtual particle moving along each lattice line exists:

$$x(t,\ \mathbf{r}) = \{x_i(t,\ \mathbf{r}) :\ x_i \in \{0,1\}\}\,, \tag{4}$$

where i indicates the velocity direction; i=1,...,6 in the 6-bit model, while i=0,...,6 in the 7-bit model. In the in the 7-bit model, a stationary particle with zero velocity is considered. For example, in the case of Fig. 1(a), the state of the lattice node at position $\mathbf{r}$ is $x_i = [1,\ 0,\ 1,\ 0,\ 1,\ 0]$ in the 6-bit model. One time evolution process consists of the collision process and propagation process. The particles moving from adjacent lattices collide with other particles at lattice nodes, and each particle changes its direction of movement according to the collision laws (Fig. 1(b)). In the collision process, particles that do not fall under the collision laws continue to move in the same direction (for example, when only one particle enters a lattice node). After the collision process, each particle moves to the adjacent lattice nodes along the lattice line at a velocity $\mathbf{c}_i$ (Fig. 1(c)). Those LGA processes can be written using Boolean variables as follows:

$$x_i(t+1,\ \mathbf{r}+\mathbf{c}_i) = x_i(t,\ \mathbf{r}) + \Delta_i[x(t,\ \mathbf{r})], \tag{5}$$

where $\Delta_i[x(t,\ \mathbf{r})]$ is the collision operator following the collision law, and the velocity vector $\mathbf{c}_i$ is given by

$$\mathbf{c}_i = \begin{cases} (0,\ 0) & (i=0) \\ \left(\cos\dfrac{\pi(i-1)}{3},\ \sin\dfrac{\pi(i-1)}{3}\right) & (i=1,\ldots,6) \end{cases}. \tag{6}$$

The collision operator holds

$$\sum_i \Delta_i(x) = 0, \quad \sum_i \mathbf{c}_i\Delta_i(x) = 0. \tag{7}$$

By Eqs. (5) and (7), we find that the collision operator satisfies the following mass and momentum conservation equations:

$$\begin{aligned} \sum_i x_i(t+1,\ \mathbf{r}+\mathbf{c}_i) &= \sum_i x_i(t,\ \mathbf{r}), \\ \sum_i \mathbf{c}_i x_i(t+1,\ \mathbf{r}+\mathbf{c}_i) &= \sum_i \mathbf{c}_i x_i(t,\ \mathbf{r}). \end{aligned} \tag{8}$$

Further details regarding the LGA models can be found in Refs. [18, 20].

3.1.2 qaLGA [16]

The collision law of LGA is constructed such that mass and momentum conservation shown in Eq. (8) is satisfied. The sum of the conservative variables at each lattice node shown in Eq. (8) is can be obtained as

$$\begin{aligned} &\text{Mass:} && \sum_i x_i(t,\,\mathbf{r}) = C^{\rho}(t,\,\mathbf{r}), \\ &\text{Momentum } (\xi\text{-direction}): && \sum_i u_i x_i(t,\,\mathbf{r}) = C^{\rho u}(t,\,\mathbf{r}), \\ &\text{Momentum } (\eta\text{-direction}): && \sum_i v_i x_i(t,\,\mathbf{r}) = C^{\rho v}(t,\,\mathbf{r}), \end{aligned} \tag{9}$$

were the momentum is decomposed into the ξ- and η-directions, and u_i and v_i are given by

$$u_i = \begin{cases} 0 \\ \cos\dfrac{\pi(i-1)}{3}, \end{cases} \quad v_i = \begin{cases} 0 & (i=0) \\ \sin\dfrac{\pi(i-1)}{3} & (i=1,\ldots,6) \end{cases}. \tag{10}$$

In general, the ξ- and η-directions correspond to the directions parallel and perpendicular to the main flow, respectively. Eq. (9) can be deformed to

$$\begin{aligned} &C^{\rho}(t,\,\mathbf{r}) - \sum_i x_i(t,\,\mathbf{r}) = 0 \\ &C^{\rho u}(t,\,\mathbf{r}) - \sum_i u_i x_i(t,\,\mathbf{r}) = 0 \\ &C^{\rho v}(t,\,\mathbf{r}) - \sum_i v_i x_i(t,\,\mathbf{r}) = 0 \end{aligned}. \tag{11}$$

Also, since the mass and momentum are conserved before and after the collision, the same relation holds even for the post-collision particle state $x_i(t',\,\mathbf{r})$, where t' is the time after the collision and before propagation:

$$\begin{aligned} &C^{\rho}(t,\,\mathbf{r}) - \sum_i x_i(t',\,\mathbf{r}) = 0 \\ &C^{\rho u}(t,\,\mathbf{r}) - \sum_i u_i x_i(t',\,\mathbf{r}) = 0 \\ &C^{\rho v}(t,\,\mathbf{r}) - \sum_i v_i x_i(t',\,\mathbf{r}) = 0 \end{aligned}. \tag{12}$$

Considering the conservation laws in Eq. (12), we define the cost function to obtain the particle states $x_i(t',\,\mathbf{r})$ after the collision process as

$$\begin{aligned} E(\{x_i(t',\,\mathbf{r})\}) = &\left[C^{\rho}(t,\,\mathbf{r}) - \sum_i x_i(t',\,\mathbf{r})\right]^2 + \left[C^{\rho u}(t,\,\mathbf{r}) - \sum_i u_i x_i(t',\,\mathbf{r})\right]^2 \\ &+ \left[C^{\rho v}(t,\,\mathbf{r}) - \sum_i v_i x_i(t',\,\mathbf{r})\right]^2 + \lambda \sum_i \left[x_i(t,\,\mathbf{r}) x_i(t',\,\mathbf{r})\right]. \end{aligned} \tag{13}$$

The first to third terms on the right-hand side of this equation correspond to the conservation equations shown in Eq. (12). In Eq. (13), those conservation equations are squared to make a quadratic function for $x_i(t',\,\mathbf{r})$. When conservation of the mass and momentum is satisfied, the first to third terms of Eq. (13) become zero. However, the state exactly the same combination as pre-collision $x_i(t,\mathbf{r})$ may also be subject to the post-collision solution since the first to third terms of Eq. (13) only consider conservation of the mass and momentum. Therefore, the fourth term is added in the proposed cost function so that the cost function is increased when the combination of x_i at time t and t' is the same. A coefficient λ is also added to the fourth term in Eq. (13). In LGA, it is necessary to follow the collision law to simulate fluid physics. However, the particles may not change their direction when the particle states are not

contained in the collision law. In those cases, the particles only propagate, maintaining their direction. For example, if only one particle enters a lattice node, no collision occurs, and the particle propagates along a lattice line without changing its direction. This property implies that the particle state x_i may not change at time t and t', which is contrary to the meaning of the fourth term of Eq. (13). Therefore, we introduce the coefficient λ to the fourth term to give a lower priority than the first to the third terms. In other words, the first to third terms must be satisfied because they are the physical conservation laws, while the fourth constraint may be broken in some cases. Expanding Eq. (13) and neglecting the constant term, we obtain the following QUBO model that reproduces the collision law of LGA:

$$E(\{x_i(t',\,\mathbf{r})\}) = \left(-2C^{\rho}\sum_i x_i'x_i' + \sum_i\sum_j x_i'x_j'\right) + \left(-2C^{\rho u}\sum_i u_i\cdot x_i'x_i' + \sum_i\sum_j u_iu_j\cdot x_i'x_j'\right) + \left(-2C^{\rho v}\sum_i v_i\cdot x_i'x_i' + \sum_i\sum_j v_iv_j\cdot x_i'x_j'\right) + \lambda\sum_i x_ix_i'x_i'. \tag{14}$$

Here for simplicity, $x_i(t,\,\mathbf{r})$ and $x_i(t',\,\mathbf{r})$ are denoted as x_i and x_i' respectively.

Figure 3 shows the velocity profile of a channel flow obtained using the QUBO model in Eq. (14), compared with that obtained by the conventional LGA. The Reynolds number based on the average velocity and the half-width δ of the channel height is Re=20, and the number of computational nodes is 55×55. The average velocity is maintained by replacing the particle at the fourth lattice node with the first lattice node with a probability of 0.37 at each time step. OpenJij [21] is used for the annealing computations. As shown in Fig. 3, the result obtained by the qaLGA is in good agreement with that of the conventional LGA, indicating that the proposed QUBO model reproduces the conventional LGA well.

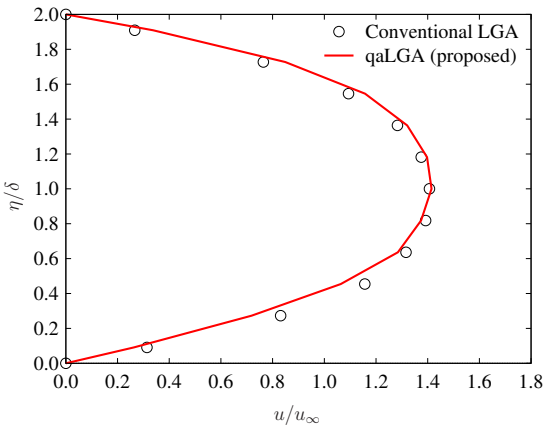


Figure 3: Velocity profiles of channel flow obtained by qaLGA and conventional LGA.

3.2 Obtaining converged flow solutions using quantum annealing

This study proposes quantum annealing computation methods that can obtain converged flow solutions by quantum annealing computation, utilizing the quantum superposition state. In the conventional numerical simulation, a converged solution is obtained using a time advancement or iterative methods from a given initial state. On the other hand, in the proposed approach, a converged solution is extracted by quantum annealing sampling from all possible solutions under the superposition state. The proposed quantum annealing computation methods are developed for LGA and finite difference methods.

The proposed QUBO model for LGA consists of the sub-cost functions as follows:

$$E(\{x\}) = E_T(\{x,x'\}) + E_\Omega(\{x,x'\}) + E_w(\{x,x'\}) + E_o(\{x\}), \tag{15}$$

where E_T, E_Ω, E_w, and E_o are the sub-cost functions for the steady solution, collision, wall boundary, and flow field conditions. Again, x' represents the particle state after the collision process (i.e., before

the propagation). For example, the sub-cost function of the collision is given by

$$E_\Omega = \sum_{\mathbf{r}\neq\mathbf{r}_{wall}} \left\{ \left(\sum_i \sum_j x'_i x'_j - \sum_i \sum_j x'_i x_j + \sum_i \sum_j x_i x_j \right) + \left(\sum_i \sum_j u_i u_j x'_i x'_j - \sum_i \sum_j u_i u_j x'_i x_j + \sum_i \sum_j u_i u_j x_i x_j \right) + \left(\sum_i \sum_j v_i v_j x'_i x'_j - \sum_i \sum_j v_i v_j x'_i x_j + \sum_i \sum_j v_i v_j x_i x_j \right) + \lambda_\omega \sum_i [x'_i x_i] \right\}, \tag{16}$$

where the idea used in Eq. (14) is extended to a whole computational domain except for the wall boundaries.

Figure 4 compares the velocity profiles of a channel flow between the proposed quantum annealing method and conventional LGA. In the conventional flow computation, the converged solution is obtained by advancing the time step from an initial condition with a constant velocity. In contrast, the proposed approach obtains the converged solution out of 10^{67} possible solutions by quantum annealing sampling. The obtained solution does not perfectly fit the solution obtained by the conventional LGA. However, the proposed approach qualitatively reproduces the velocity profile in which the flow velocity decelerates near the walls and accelerates near the center of the channel.

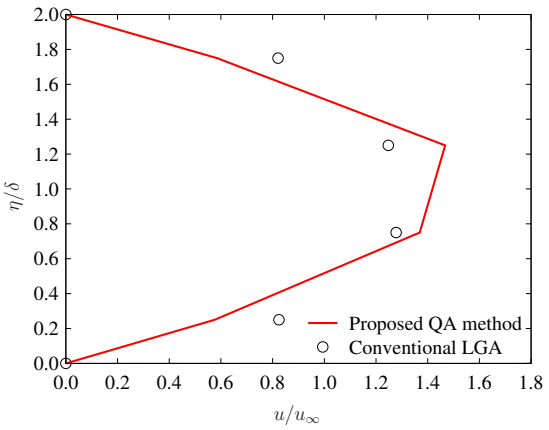


Figure 4: Velocity profiles of channel flow obtained by proposed quantum annealing method and conventional LGA. The proposed quantum annealing method obtains the converged velocity profile by solving the cost function based on LGA discretization.

Equation (15) describes the cost function based on LGA discretization to obtain a converged flow solution by quantum annealing. Similarly, a cost function to obtain a converged flow solution can be built for finite difference methods. Figure 5 compares the velocity profiles of a channel flow obtained by the proposed quantum annealing method and analytical solution. In the proposed method, the cost function based on finite difference discretization is solved by quantum annealing to obtain the converged solution. The proposed approach obtains the converged solution out of 10^{14} possible solutions by quantum annealing. The obtained solution is in good agreement with the analytical solution.

4 Conclusions

This study has proposed numerical methods to obtain the flow solutions by quantum annealing. Quantum annealing has attracted attention in recent years as a faster and/or more accurate method for finding optimal solutions in combinatorial optimization problems than conventional methods. This study first introduces the quantum annealing lattice gas automata (qaLGA) proposed by a recent paper. In a numerical experiment, the qaLGA well reproduces the velocity profile of a channel flow obtained by the

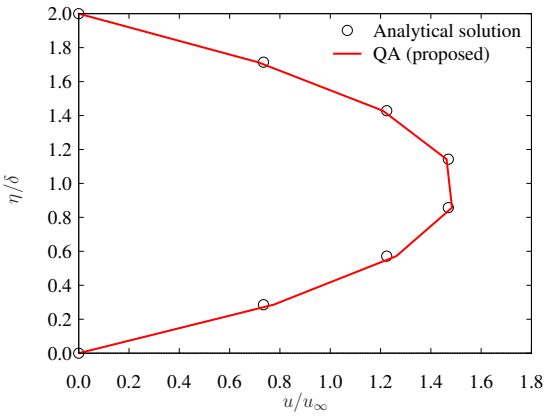


Figure 5: Velocity profile of a channel flow obtained by proposed algorithm using finite difference methods and analytical solution.

conventional LGA. This study then proposes numerical methods to obtain converged flow solutions by annealing computation, utilizing the quantum superposition state. The proposed methods are built for lattice gas automata (LGA) and finite difference methods. In numerical experiments, both proposed methods obtain the velocity profiles of a channel flow from a number of possible solutions using quantum annealing sampling. In particular, the velocity profile obtained by the proposed quantum annealing method based on finite difference methods agrees well with the analytical solution.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP22K04530.

References

[1] Hiroyuki Asada, Yoshiharu Tamaki, Ryoji Takaki, Takaaki Yumitori, Shun Tamura, Keita Hatanaka, Kazuhiro Imai, Hirotaka Maeyama, and Soshi Kawai. FFVHC-ACE: fully automated Cartesian-grid-based solver for compressible large-eddy simulation. *AIAA Journal*, 61(8):3466–3484, 2023.

[2] Yuichi Kuya, Kosuke Totani, and Soshi Kawai. Kinetic energy and entropy preserving schemes for compressible flows by split convective forms. *Journal of Computational Physics*, 375:823–853, 2018.

[3] Yuichi Kuya and Soshi Kawai. A stable and non-dissipative kinetic energy and entropy preserving (keep) scheme for non-conforming block boundaries on cartesian grids. *Computers & Fluids*, 200:104427, 2020.

[4] Yoshiharu Tamaki and Soshi Kawai. Wall modeling for large-eddy simulation on non-body-conforming Cartesian grids. *Physical Review Fluids*, 6(11):114603, 2021.

[5] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[6] Jeffrey Yepez. Quantum lattice-gas model for computational fluid dynamics. *Physical Review E*, 63(4):046702, 2001.

[7] Yudong Cao, Anargyros Papageorgiou, Iasonas Petras, Joseph Traub, and Sabre Kais. Quantum algorithm and circuit design solving the Poisson equation. *New Journal of Physics*, 15(1):013021, 2013.

[8] René Steijl and George N. Barakos. Parallel evaluation of quantum algorithms for computational fluid dynamics. *Computers & Fluids*, 173:22–28, 2018.

[9] Guanglei Xu, Andrew J Daley, Peyman Givi, and Rolando D Somma. Turbulent mixing simulation via a quantum algorithm. *Aiaa Journal*, 56(2):687–699, 2018.

[10] Frank Gaitan. Finding flows of a Navier–Stokes fluid through quantum computing. *npj Quantum Information*, 6(1):1–6, 2020.

[11] Frank Gaitan. Finding solutions of the Navier–Stokes equations through quantum computing–recent progress, a generalization, and next steps forward. *Advanced Quantum Technologies*, 4(10):2100055, 2021.

[12] Dieter Jaksch, Peyman Givi, Andrew J Daley, and Thomas Rung. Variational quantum algorithms for computational fluid dynamics. *AIAA journal*, 61(5):1885–1894, 2023.
[13] Wael Itani, Katepalli R Sreenivasan, and Sauro Succi. Quantum algorithm for lattice Boltzmann (QALB) simulation of incompressible fluids with a nonlinear collision term. *Physics of Fluids*, 36(1), 2024.
[14] D-Wave Systems, Advantage, https://www.dwavesys.com/solutions-and-products/systems, accessed 2024-06-30.
[15] Navamita Ray, Tirtha Banerjee, Balu Nadiga, and Satish Karra. On the viability of quantum annealers to solve fluid flows. *Frontiers in Mechanical Engineering*, page 68, 2022.
[16] Yuichi Kuya, Kazuhiko Komatsu, Kouki Yonaga, and Hiroaki Kobayashi. Quantum annealing-based algorithm for lattice gas automata. *Computers & Fluids*, 274:106238, 2024.
[17] U. Frisch, B. Hasslacher, and Y. Pomeau. Lattice–gas automata for the Navier–Stokes equation. *Phys. Rev. Lett.*, 56:1505–1508, Apr 1986.
[18] Dominique d'Humieres and Pierre Lallemand. Numerical simulations of hydrodynamics with lattice gas automata in two dimensions. *Complex Systems*, 1:599–632, 1987.
[19] Uriel Frisch, Dominique d'Humières, Brosl Hasslacher, Pierre Lallemand, Yves Pomeau, and Jean-Pierre Rivet. Lattice gas hydrodynamics in two and three dimensions. *Complex Systems*, 1:649–707, 1987.
[20] Dieter A Wolf-Gladrow. *Lattice-gas cellular automata and lattice Boltzmann models: an introduction*. Springer, 2004.
[21] OpenJij, https://www.openjij.org, accessed 2024-06-30.