Quantum Information Processing (2022) 21: 322
https://doi.org/10.1007/s11128-022-03667-7

Variational quantum solutions to the advection–diffusion equation for applications in fluid dynamics

Reuben Demirdjian[1] · Daniel Gunlycke[2] · Carolyn A. Reynolds[3] · James D. Doyle[3] · Sergio Tafur[4]

Received: 22 February 2022 / Accepted: 25 August 2022 / Published online: 17 September 2022

Abstract
Constraints in power consumption and computational power limit the skill of operational numerical weather prediction by classical computing methods. Quantum computing could potentially address both of these challenges. Herein, we present one method to perform fluid dynamics calculations that takes advantage of quantum computing. This hybrid quantum–classical method, which combines several algorithms, scales logarithmically with the dimension of the vector space and quadratically with the number of nonzero terms in the linear combination of unitary operators that specifies the linear operator describing the system of interest. As a demonstration, we apply our method to solve the advection–diffusion equation for a small system using IBM quantum computers. We find that reliable solutions of the equation can be obtained on even the noisy quantum computers available today. This and other methods that exploit quantum computers could replace some of our traditional methods in numerical weather prediction as quantum hardware continues to improve.

Keywords Earth and atmospheric sciences · Partial differential equations · Linear system of equations · Fluid mechanics

1 Introduction

Numerical weather prediction (NWP) is a subset of the broader field of fluid dynamics that seeks to provide solutions to systems of partial differential equations (PDEs),

✉ Reuben Demirdjian
Reuben.Demirdjian.ctr@nrlmry.navy.mil

[1] National Research Council, Monterey, CA 93943, USA

[2] Naval Research Laboratory, Washington, DC 20375, USA

[3] Naval Research Laboratory, Monterey, CA 93943, USA

[4] NavAir Warfare Centers –Aircraft Division, Patuxent River, MD 20770, USA

namely the Navier–Stokes equations, ultimately amounting to solving an initial value problem. One challenge in NWP is that the spatial scales for a forecast can range from molecular to planetary scale (10^7 m), sometimes called the "tyranny of scales" [1], resulting in a continual drive toward creating operational weather models with finer and finer grid spacing. Finer grids in turn require larger calculations with increased power demands. With current model designs, downscaling an operational model from 10 to 1 km horizontal resolution would increase the power consumption by approximately three orders of magnitude [2]. Furthermore, "resolutions of 1 – 5 km ... are crucial for resolving convection" and with current model designs, a "high-performance computer of unprecedented dimension and cost would be required" to perform operational simulations at those resolutions [2]. This suggests that operational NWP is approaching a power consumption limit that would halt resolution-based forecasting skill improvements [3].

Quantum Computing (QC) has the potential to alleviate the power consumption roadblock faced in NWP. For example, the estimated energy consumption to perform Google's recent quantum supremacy calculation on a quantum computer [4] was recently compared with that of Summit, the world's most powerful supercomputer [5, 6]. It was found that the energy consumptions of Summit and the quantum computer were 780,000 kWh and 1.4 kWh, respectively [5]. There are two main reasons for the 5.7 orders of magnitude ratio: (i) the calculation on the quantum computer took far less time to run, 200 s vs 2.5 days [7], and (ii) the power consumption of the single-chip quantum computer is far less than that of Summit, which has roughly the size of two tennis courts [6].

Another challenge that NWP faces is the end of Moore's law [8] — the observation that computing power doubles approximately every 18 months due to the increase in transistor chip density. Transistor size cannot shrink indefinitely and while there is uncertainty on when the limit will be reached, some studies suggest that could be as soon as 2030 [9]. Regardless of the precise timing, Moore's law coming to an end will directly impact operational NWP. Capping computational power will impede forecast skill improvements by limiting several modeling facets such as parameterization methods, data assimilation, and further downscaling of model resolution.

The challenges posed by the increasing power consumption and the end of Moore's law do not necessarily imply the end of forecast skill improvements, as there are several other promising approaches that could improve upon current model design. Some potential methodologies include improved parameterization theories, new model frameworks like machine learning [10–12], mathematically solving the governing equations of fluid dynamics [13], or, as focused on in this study, the exploitation of QC. [14]

Recently, there has been a flurry of publications exploring the possibility to efficiently solve partial differential equations, which are directly relevant for NWP, using QC [14–35]. Some of these algorithms are developed for long-term applications relying on fault-tolerant QC, others for nearer-term applications using Noisy Intermediate Scale Quantum (NISQ) computers. Although the number of gate operations in NISQ algorithms are limited owing to quantum decoherence in NISQ computers, they can still offer a computational advantages over their classical counterparts. A prominent example is the variational style algorithm [36] that uses a hybrid quantum–classical approach to leverage the advantages both have to offer. The present study applies a variational approach to investigate what is required of NISQ algorithms to offer a computational advantage for NWP applications.

2 Methods

2.1 The advection–diffusion equation

The advection–diffusion equation, a generalization of the Burgers equation, is a nonlinear PDE that describes the change in a quantity undergoing advection and diffusion. In the case where the quantity of interest is momentum, the one-dimensional form of this equation is given by

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} + \nu\frac{\partial^2 u}{\partial x^2}, \tag{1}$$

where u is velocity, t is time, x is the spatial coordinate, and ν is the diffusion coefficient, herein set to 5×10^{-2}. This equation is an example of a fundamental initial value problem in fluid dynamics. While it does not have any direct application in operational NWP, solving this simplified equation using QC as a proof-of-concept is an initial step toward solving equations with increased complexity.

2.2 Advection–diffusion matrix equation for quantum systems

A quantum algorithm is applied that solves the initial value problem for dissipative quadratic PDEs through use of the forward Euler method to set up a linear system of equations [22]. This specific class of problems includes several equations relevant in fluid dynamics, e.g., the advection–diffusion equation and Navier–Stokes equations. One issue in solving nonlinear equations using QC is the fact that quantum mechanics is fundamentally a linear theory. To overcome this challenge, a linearization method is performed that introduces powers of the variables in the nonlinear differential equation and maps the problem to an infinite sequence of coupled linear differential equations [22]. An approximation of the system of equations is made by truncating the infinite series, whereby including more terms will increase the accuracy of solution.

The vector space dimension N of the linear system of equations for this method is given by

$$N = N_T n\frac{n^\tau - 1}{n - 1}, \tag{2}$$

where n is the number of nonlinear differential equations, τ is the truncation order, and N_T is the number of discrete times, including zero (cf. Ref. [22]). Although the dimension N grows exponentially with τ, the accuracy of the solution also "converges " exponentially with τ for the advection–diffusion equation, allowing us to choose τ small, e.g., $\tau \leq 5$ [22]. Since a low truncation order can be safely used, the most

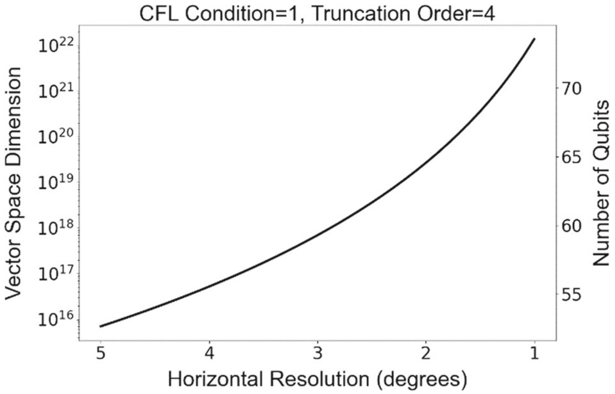

Fig. 1 The vector space dimension required to solve the advection–diffusion equation using the methods discussed in this paper as a function of the horizontal resolution of a two-dimensional 10-day global forecasting model. The right y-axis shows the number of qubits required to process a corresponding Hilbert space of equal or greater dimension on a quantum computer

important variable for the size of N is n, which in NWP is the number of prognostic variables multiplied by the number of horizontal and vertical grid cells.

A rough estimate is performed using (2) to determine if solving a global NWP model using QC is a feasible prospect. Consider the model representing a 10-day global forecast with a 5-degree horizontal resolution, 1 vertical level, and 5 prognostic variables. Assuming that $u_{max} = 100$ m/s and the Courant-Friedrichs-Lewy (CFL) condition is set to 1 for numeric stability, then $\Delta t = 5574$ s, which amounts to $N_T = 156$ total times. From (2), it follows that this requires a vector space dimension on the order of 10^{15}. Moreover, assuming an efficient state preparation method exists, then a quantum computer would require 52 low-noise qubits to build and solve this linear system of equations, a prospect that is within the realm of possibility within a decade [37]. Note that this vector space dimension is not the same used in a conventional NWP model on a classical computer to solve this type of problem because the quantum algorithm requires a linearization of the terms resulting in additional variables [22]. Figure 1 repeats this calculation by varying the horizontal resolution but keeping the CFL condition constant. While the vector space dimension ranges from 10^{15} to 10^{22}, the corresponding number of required qubits only ranges from 52 to 73. This illustrates that exponentially large problems could in principle be solved using only a linear amount of QC resources. Furthermore, it suggests that the vector space dimension of the linear system of equations is not the primary obstacle for solving PDEs on a QC. A discussion about other obstacles that must be overcome in order to solve large fluid dynamics problems is provided in Sect. 4.

2.3 2.3. Matrix decomposition

To be able to perform QC operations, we must first map the computational vector space to the Hilbert space of a quantum register inside a quantum computer. As this quantum register comprises qubits, the Hilbert space is a tensor power of two-dimensional Hilbert spaces of the qubits in the register. Furthermore, we require that all the states of these Hilbert spaces are unit normalized, which implies that all QC operations must be unitary. This requirement is automatically met if we decompose the linear operator describing the system of interest as

$$A = \sum_{l=1}^{L} c_l A_l, \tag{3}$$

where α_l is a complex coefficient, A_l is a unitary operator, and L is the number of nonzero coefficients in the decomposition. Each unitary operator A_l is given by

$$A_l = \otimes_q \sigma_{lq}, \tag{4}$$

where each σ_{lq} is either the two-dimensional identity operator I_q or one of the Pauli operators X_q, Y_q, and Z_q acting on the states of qubit q in the register. For an N x N matrix, the decomposition can be performed as a preprocessing step involving between $O(N)$ and $O(N^2 \log_2 N)$ operations depending on the sparsity of the operator matrix [38].

2.4 Quantum solutions to linear system of equations

While the most advantageous linear system of equation solvers are out of reach for NISQ hardware, the Variational Quantum Linear Solver (VQLS) algorithm can theoretically provide an advantage over classical computers for large systems when solving the $A|x\rangle = |b\rangle$ problem [39]. This algorithm uses the popular quantum–classical variational hybrid method [36] to solve for $|x\rangle$ leveraging the fact that a quantum computer can efficiently determine the expectation value of some observable, while the classical computer can use this expectation value to calculate a cost function. In practice, we first prepare a random initial parameter vector$\boldsymbol{\theta} = \boldsymbol{\theta}^{\boldsymbol{init}}$, where $\boldsymbol{\theta} = (\theta_0, \theta_1, \ldots, \theta_P)$ consists of P components of rotation angles between 0 and2π. The vector $\boldsymbol{\theta}$ specifies the quantum state$|x(\boldsymbol{\theta})\rangle = V(\boldsymbol{\theta})|0\rangle$, where $V(\boldsymbol{\theta})$ describes the quantum circuit (or ansatz), and $|0\rangle$ is the initially prepared quantum state on the register Hilbert space. The algorithm then iterates over the parameter vector$\boldsymbol{\theta}$, which is updated to minimize a cost function $C(\boldsymbol{\theta})$ that describes how "close" $A|x(\boldsymbol{\theta})\rangle$ is to$|b\rangle$. Optimization of the cost function occurs when $|x(\boldsymbol{\theta})\rangle \approx |x\rangle$ to within some termination condition.

The local cost function discussed in [39] and applied by both [40, 41] is used, which avoids the barren plateau issue discussed in [42], and converges on the global minima regardless of the starting point. This is given by

$$C(\boldsymbol{\theta}) = \frac{\langle x(\boldsymbol{\theta})|H|x(\boldsymbol{\theta})\rangle}{\langle \psi(\boldsymbol{\theta})|\psi(\boldsymbol{\theta})\rangle}, \tag{5}$$

where $|\psi(\boldsymbol{\theta})\rangle = A|x(\boldsymbol{\theta})\rangle$ and

$$H = A^{\dagger}U\left(I - \frac{1}{Q}\sum_{q=1}^{Q}|0_q 0_q| \otimes I_{\hat{q}}\right)U^{\dagger}A, \tag{6}$$

where the unitary operator U prepares $|b\rangle$ such that $U|0\rangle = |b\rangle$, Q is the number of qubits in the register, and $I_{\hat{q}}$ is the identity operator for all qubits except q. Substituting Eq. (3) and (6) in Eq. (5), and using the identity $|0_q\rangle\langle 0_q| = (I_q + Z_q)/2$ yields

$$C(\boldsymbol{\theta}) = \frac{1}{2} - \frac{1}{2Q}\left(\frac{\sum_{q=1}^{Q}\sum_{ll'} c_l c_{l'}^{*}\delta_{ll'}^{q}}{\sum_{ll'} c_l c_{l'}^{*}\beta_{ll'}}\right), \tag{7}$$

where

$$\delta_{ll'}^{q} = x(\theta)|A_{l'}^{\dagger}UZ_qU^{\dagger}A_l|x(\theta), \tag{8}$$

and

$$\beta_{ll'} = x(\theta)|A_{l'}^{\dagger}A_l|x(\theta). \tag{9}$$

Herein, we use a modified version of the ansatz circuit 9 from [43] for $V(\boldsymbol{\theta})$, as shown in Fig. 2 a. This ansatz has been selected to constrain the solution space to include only real solutions. The circuit U that prepares $|b\rangle$ is shown in Fig. 2 b, where the angle ϕ is determined from the initial condition of $u(x, t = 0)$ discussed in Sect. 3.

One key difference in the implementation of the VQLS method herein versus that in Ref. [39] is that the present study does not use the Hadamard test or Hadamard-overlap test methods to estimate the expectation values. While those methods have the advantage of only needing to measure the ancillary qubit, they also have the disadvantage of requiring that all unitary matrices from the linear combination of unitaries method be controlled to the ancillary qubit, thereby adding substantial two-qubit gate noise. Instead, the QISKIT [41] Pauli expectation method is used that converts the operator into Pauli strings, diagonalizes the ansatz circuit in each basis

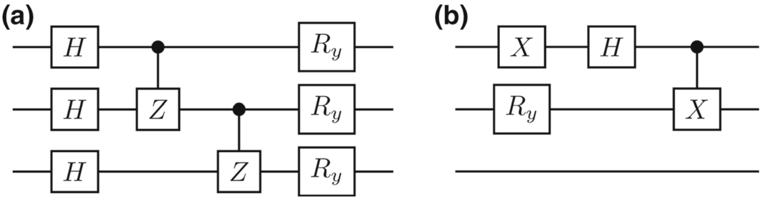

Fig. 2 **a** A unit of the ansatz that is repeated in the full quantum circuit depending on the specified circuit depth. H is the Hadamard gate, Z is the Pauli Z-gate, and $\mathrm{R_y}$ is the rotation gate about the y-axis for angle θ discussed in text. **b** The quantum circuit used to prepare the b-vector. X is the Pauli X-gate, and the angle ϕ is used to rotate $\mathrm{R_y}$

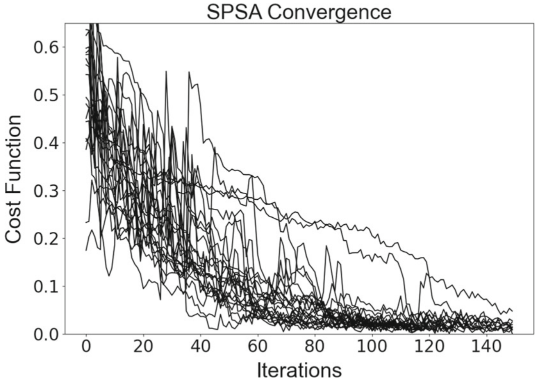

Fig. 3 The local cost function as a function of the iteration using the SPSA optimization method for the 24 real QC runs

of the Pauli strings, and finally measures every qubit. The advantage of this method is that it reduces the required number of two-qubit gates. The consequence is that each qubit must be measured individually. This method is appropriate for small circuits but future implementations of this algorithm should consider both the 2-qubit gate noise and readout assignment error when deciding between the discussed methods.

There are several optimizers that could be applied to implement the VQLS method. Herein, we use the Stochastic Perturbation Simultaneous Approximation (SPSA) method [44, 45]. This particular optimizer requires fewer function evaluations to estimate the gradient than other methods, which saves valuable quantum computing resources. The chosen hyper-parameters, in the notation of Refs. [44] and [45], are: $\alpha = 0.602, \gamma = 0.101, A = 10, a = 4, c = 0.1$. Convergence is determined when 5 successive iterations have a tolerance below 2×10^{-2}. The SPSA optimizer is initialized with random starting locations within the domain during each run of the VQLS algorithm. The convergences of the 24 runs performed in this study are shown in Fig. 3 where rapid convergence toward zero is observed in the first 40 iterations with slower convergence thereafter.

2.5 IBMQ Hardware

The VQLS algorithm was run on three different IBM-Quantum [41] systems: (i) Cairo v1.0.2 – a 27 qubit system with a quantum volume [46] of 64 using the Falcon r5.11 processor, (ii) Hanoi v1.0.2 – a 27 qubit system with a quantum volume of 64 using the Falcon r5.11 processor, and (iii) Montreal v1.10.5 – a 27 qubit system with a quantum volume of 128 using the Falcon r4 processor. Table 1 lists an instance of the calibration data around the time the calculations herein were performed.

3 Results

The one-dimensional advection–diffusion equation is discretized to a stencil with $n = 4$ and initial conditions $u(x, t = 0) = \sin(\kappa x)$ where $x\varepsilon[0, L_D]$, $\kappa = 2\pi/L_D$, and $L_D = 1$ is the spatial domain size. With these initial conditions, the analytic

Table 1 An instance of calibration data of the three IBM quantum computers used in this study obtained at the time the calculations herein were performed

	Pauli-X	Average CNOT	Readout	T1 (ms)	T2 (ms)	Readout Length (ns)
Cairo	3e-4	1e-2	1.3e-2	90	100	730
Hanoi	2e-4	1e-2	1.5e-2	130	120	750
Montreal	5e-4	1e-2	3e-2	100	75	5000

solution to Eq. (1) is given by $u(x,t) = e^{-\kappa \nu t}\sin(\kappa x)$. Following the method in Ref. [22], the operator A in the linear system of equations $A|x\rangle = |b\rangle$ for $N_T = 3$ is represented by a matrix of the form

$$A = \begin{pmatrix} I & & \\ -(I + M\Delta t) & I & \\ & -(I + M\Delta t) & I \end{pmatrix}, \tag{10}$$

where I is a unit matrix, M is a matrix describing the system, and Δt is a time step size. Linearizing our system, i.e., letting $\tau = 1$, we find that

$$M = \begin{pmatrix} -0.9 & 0.45 & 0 & 0.45 \\ 0.45 & -0.9 & 0.45 & 0 \\ 0 & 0.45 & -0.9 & 0.45 \\ 0.45 & 0 & 0.45 & -0.9 \end{pmatrix}, \tag{11}$$

and $\Delta t = 0.25s$.

Since the first block row of A contains only the identity, it is straightforward to reduce the dimension of our computational vector space from $N = N_T n = 12$ to $N = 8$. This reduction transforms the state $|b\rangle$ to the form $|b\rangle = \frac{1}{\sqrt{2}}\left(\cos\frac{\phi}{2}, -\sin\frac{\phi}{2}, \sin\frac{\phi}{2}, -\cos\frac{\phi}{2}, 0, 0, 0, 0\right)^T$ where $\phi = -160.725$ follows from our initial conditions. Our reduced description allows us to bijectively map the computational space to the Hilbert space of a quantum register containing $Q = 3$ qubits, as the dimension of this Hilbert space is $2^Q = 8$, i.e., the same as the computational space.

To demonstrate a proof of concept, a 24-member ensemble of the quantum solution for the advection–diffusion equation is produced on IBM's Cairo, Hanoi, and Montreal machines using 3 of their 27 available qubits and implementing the maximum 8,192 shots for each circuit. Four units of the circuit in Fig. 2a are used to form the ansatz. Each quantum state generated by this ansatz is uniquely specified by a parameter vector comprising 12 components, one for each R_y gate. The cost function per iteration is shown in Fig. 3 for the 24 simulations. Most trajectories are qualitatively similar and approach a cost function value approximately 10^{-2}.

Three different types of solutions to the advection–diffusion equation are shown in Fig. 4: (i) the analytic solution described in Sect. 2.1, (ii) a solution to the matrix equation $Ax = b$ computed classically, and (iii) the solution to $A|x\rangle = |b\rangle$ obtained

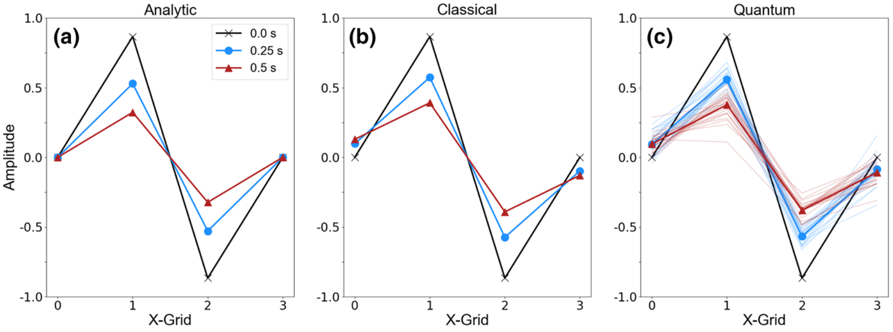

Fig. 4 Solutions to the advection–diffusion equation at three time steps for the **a** analytic solution, **b** classical solution of the linear system of equations discussed in the text, **c** same as (**b**) except for the quantum solutions of the 24 different runs on IBM's quantum hardware (thin lines) and average of the 24 runs (thick lines). For all plots the initial condition is shown in black, $0.25s$ time step in blue, and $0.5s$ time step in red

using the VQLS method. Note that if the quantum solution (iii) was perfect, it would reproduce the classical solution in (ii) and not the analytic solution in (i), as the linear system of equations are themselves only approximate solutions to (1). However, a comparison of Figs. 4 a and b shows that even for the truncation order $N = 1$ [22] the linear system of equations produce adequate solutions for our purposes. Qualitative comparisons of the average of all 24 quantum solutions (bold lines in Fig. 4 c) and the classical solutions (Fig. 4 b) clearly demonstrate that reasonable solutions can be obtained using the VQLS method. The root mean square error of the 24-solution average at $t = 0.25$ s is 0.010m/s and at $t = 0.5$ s is 0.021m/s, corresponding to 6% and 15% relative errors.

While the mean quantum solution (Fig. 4c) looks nearly identical to that of the classical solution (Fig. 4b), there is a clear spread in the individual solutions. It is therefore logical to ask: do the individual solutions contain biases as the optimizer iterations tend toward infinity? In other words, what is the sensitivity of the solution accuracy to the stopping iteration? To investigate this, an arbitrary run is selected where the solution at each grid point is shown in Supplemental Figs. 1 and 2 for $t = 0.25$ s and $t = 0.5$ s, respectively. The supplemental figures illustrate that only 1 out of the 8 solutions do *not* have an inherent bias after convergence, while 7 of 8 do exhibit some kind of bias. This implies that the sign of the errors in each of the solutions may not be randomly based on which iteration the solution happens to be stopped on, but rather that after reasonable convergence the sign of the error is already determined. This might necessitate multiple integrations for an unbiased estimate.

4 Discussion and Conclusions

This study is focused on combining several novel algorithms to (i) set up a nonlinear partial differential equation into a linear system of equations [22], (ii) perform a Pauli decomposition on the resulting matrix equation [38], and (iii) solve the decomposed linear system of equations on IBM's quantum hardware using the VQLS method

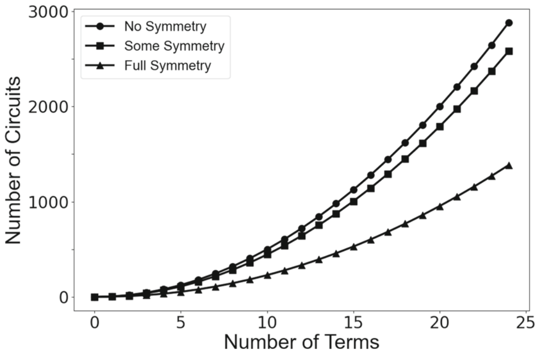

Fig. 5 The number of circuits required as a function of the number of terms in the Pauli decomposition of the matrix A. Three cases are shown with various amounts of circuit symmetry used to reduce the total number of circuits (see text for discussion)

[39]. While this investigation is not the first to solve partial differential equations on real hardware [21, 23], it is the first to solve a meteorologically relevant computational problem that demonstrates the feasibility for quantum computers to become a potential tool for numerical weather prediction.

The question remains: can the methods applied in this study be scaled up to solve much larger systems of equations and thereby build more sophisticated fluid models? In some particular cases it is possible, however, this algorithm cannot be scaled up to solve arbitrary problems in its current form. The bottleneck in the workflow is the cost function in the VQLS algorithm that can require an enormous number of circuits as the number of terms in the linear combination of unitaries grows. As there are QL^2 and L^2 coefficients $\beta_{ll}^{'}$ and $\delta_{ll'}^{q}$ in Eq. (8) and (9), respectively, the number of circuits required for each iteration of the VQLS is

$$N_C = (Q + 1)L^2. \tag{12}$$

Some symmetries are exposed in the $\beta_{ll'}$ term by observing that $\beta_{ll'} = 1$ when $l = l^{'}$ and that $\beta_{ll'} = \beta_{l'l}$, bringing a small reduction in the number of circuits as shown in Fig. 5 [39]. Similar symmetries were found to empirically exist for the $\delta_{ll'}^{q}$ term that allow for an even greater reduction by observing that $\delta_{l'l}^{q} = \delta_{ll'}^{q}$ reducing the overall number of circuits by about a factor of two from the baseline case (Fig. 5). It is important to note that the latter symmetry was found empirically to work for the specific case outlined in this study and that it is not in general true. A useful point of reference for Fig. 5 is that the current maximum N_C that can be submitted to IBM's quantum systems is 900. Therefore, any algorithm that requires an N_C significantly greater than 900 cannot be practically implemented at this time. Even in the full symmetry case, the quadratic relationship prevents the methods discussed in this paper from being scaled up substantially since N_c quickly becomes too large to be realistically run on a QC.

The specific type of case in which the VQLS can be implemented is therefore when N_C is relatively small. It is unlikely that an arbitrary matrix will decompose into a small number of unitaries, so unless care is taken to prepare a specific matrix, the cost function for the VQLS algorithm will require some improvements to be useful for solving nonlinear differential equations in form investigated here. A current challenge is therefore to determine if the N_C scaling can be improved upon thereby opening the doors for the VQLS algorithm to be potentially useful for NWP studies on a larger scale. However, this is not a long-term challenge of fault-tolerant systems because in the future we will be utilizing completely different algorithms to solve this type of problem.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11128-022-03667-7.

Acknowledgements RD, DG, CAR, and JDD has been supported by the Office of Naval Research (ONR) through the NRL Base Program, PE 0601153N. We acknowledge quantum computing resources from IBM through a collaboration with the Air Force Research Laboratory (AFRL).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Data availability statement Due to confidentiality agreements, supporting data can only be made available to bona fide researchers subject to a non-disclosure agreement. Details of the data and how to request access are available from Reuben Demirdjian at the U.S. Naval Research Laboratory.

References

1. J. T. Oden, T. Belytschko, J. Fish, T. J.R. Hughes, C. Johnshon, D. Keyes, A. Laub, L. Petzold, D. Srolovitz, S. Yip and J. Bass, "Report of the National Science Foundation Blue Ribbon Panel on Simulation-Based Engineering Science," (2006)
2. Bauer, P., Thorpe, A., Brunet, G.: The quiet revolution of numerical weather prediction. Nature **525**, 47–55 (2015)
3. R. Loft, "Earth System Modeling Must Become More Energy Efficient," *Eos,* 28 (2020)
4. Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J.C., Barends, R., Biswas, R., Boixo, S., Brandao, F.G.S.L., Buell, D.A.: Quantum supremacy using a programmable superconducting processor. Nature **574**, 505–510 (2019)
5. C. Berger, A. Di Paolo, T. Forrest, S. Hadfield, N. Sawaya, M. Stęchły and K. Thibault, "Quantum technologies for climate change: Preliminary assessment," *arXiv preprint* arXiv:2107.05362*,* (2021)
6. J. Preskill, "Quantum computing 40 years later," *arXiv preprint* arXiv:2106.10522*,* (2021)
7. E. Pednault, J. Gunnels, D. Maslov and J. Gambetta, "On quantum supremacy," IBM Research Blog, vol. 21, (2019)
8. Schaller, R.R.: Moore's law: past, present and future. IEEE Spectr. **34**, 52–59 (1997)
9. Burg, D., Ausubel, J.H.: Moore's Law revisited through Intel chip density. PLoS ONE **16**, e0256245 (2021)
10. Schultz, M.G., Betancourt, C., Gong, B., Kleinert, F., Langguth, M., Leufen, L.H., Mozaffari, A., Stadtler, S.: Can deep learning beat numerical weather prediction? Phil. Trans. R. Soc. A **379**, 20200097 (2021)
11. Weyn, J.A., Durran, D.R., Caruana, R.: Can machines learn to predict weather? Using deep learning to predict gridded 500-hPa geopotential height from historical weather data. J. Adv. Model. Earth Syst. **11**, 2680–2693 (2019)
12. J. A. Weyn, D. R. Durran and R. Caruana, "Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere," *Journal of Advances in Modeling Earth Systems,* vol. 12, p. e2020MS002109, (2020)
13. "Clay Math Institute," [Online]. Available: http://www.claymath.org/millennium-problems/navier%E2%80%93stokes-equation.

14. F. Gaitan, "Finding flows of a Navier–Stokes fluid through quantum computing," *Quantum Information Processing* , vol. 6, no. 1, (2020)
15. F. Oz, R. K. S. S. Vuppala, K. Kara and F. Gaitan, "Solving Burgers' equation with quantum computing," *Quantum Information Processing,* vol. 21, no. 1, (2022)
16. P. García-Molina, J. Rodríguez-Mediavilla and J. J. García-Ripoll, "Solving partial differential equations in quantum computers," *arXiv preprint* arXiv:2104.02668*,* 2021.
17. Engel, A., Smith, G., Parker, S.E.: Linear embedding of nonlinear dynamical systems and prospects for efficient quantum algorithms. Phys. Plasmas **28**, 062305 (2021)
18. W. Itani, "Fluid Dynamicists Need to Add Quantum Mechanics into their Toolbox," (2021)
19. Berry, D.W.: High-order quantum algorithm for solving linear differential equations. J. Phys. A: Math. Theor. **47**, 105301 (2014)
20. Berry, D.W., Childs, A.M., Ostrander, A., Wang, G.: Quantum algorithm for linear differential equations with exponentially improved dependence on precision. Commun. Math. Phys. **356**, 1057–1081 (2017)
21. Xin, T., Wei, S., Cui, J., Xiao, J., Arrazola, I., Lamata, L., Kong, X., Lu, D., Solano, E., Long, G.: Quantum algorithm for solving linear differential equations: theory and experiment. Phys. Rev. A **101**, 032307 (2020)
22. J.-P. Liu, H. Ø. Kolden, H. K. Krovi, N. F. Loureiro, K. Trivisa and A. M. Childs, "Efficient quantum algorithm for dissipative nonlinear differential equations," Proceed. National Acad. Sci., vol. 118, (2021)
23. Suau, A., Staffelbach, G., Calandra, H.: Practical Quantum Computing: solving the wave equation using a quantum approach. ACM Trans. Quantum Comput. **2**, 1–35 (2021)
24. Arrazola, J.M., Kalajdzievski, T., Weedbrook, C., Lloyd, S.: Quantum algorithm for nonhomogeneous linear partial differential equations. Phys. Rev. A **100**, 032306 (2019)
25. S. K. Leyton and T. J. Osborne, "A quantum algorithm to solve nonlinear differential equations," *arXiv preprint* arXiv:0812.4423*,* (2008)
26. Childs, A.M., Liu, J.-P.: Quantum spectral methods for differential equations. Commun. Math. Phys. **375**, 1427–1457 (2020)
27. A. M. Childs, J.-P. Liu and A. Ostrander, "High-precision quantum algorithms for partial differential equations," *arXiv preprint* arXiv:2002.07868*,* (2020)
28. Costa, P.C.S., Jordan, S., Ostrander, A.: Quantum algorithm for simulating the wave equation. Phys. Rev. A **99**, 012323 (2019)
29. S. Lloyd, G. De Palma, C. Gokler, B. Kiani, Z.-W. Liu, M. Marvian, F. Tennie and T. Palmer, "Quantum algorithm for nonlinear differential equations," *arXiv preprint* arXiv:2011.06571*,* (2020)
30. Kyriienko, O., Paine, A.E., Elfving, V.E.: Solving nonlinear differential equations with differentiable quantum circuits. Phys. Rev. A **103**, 052416 (2021)
31. Childs, A.M., Kothari, R., Somma, R.D.: Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. SIAM J. Comput. **46**, 1920–1950 (2017)
32. Lubasch, M., Joo, J., Moinier, P., Kiffner, M., Jaksch, D.: "Variational quantum algorithms for nonlinear problems," Physical Review A, **101**, p. 010301 (2020)
33. Gourianov, N., Lubasch, M., Dolgov, S.: "A quantum-inspired approach to exploit turbulence structure," Nat Comput Sci, **2**, p. 30–37 (2022). https://doi.org/10.1038/s43588-021-00181-1
34. Budinski, L.: "Quantum algorithm for the advection–diffusion equation simulated with the lattice Boltzmann method," Quantum Inf Process, **20**, p. 57 (2021). https://doi.org/10.1007/s11128-021-02996-3
35. Budinski, L.: "Quantum algorithm for the Navier–Stokes equations by using the streamfunction-vorticity formulation and the lattice Boltzmann method," International Journal of Quantum Information, **20**, p. 2150039 (2022). https://doi.org/10.1142/S0219749921500398
36. Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P.J., Aspuru-Guzik, A., O'brien, J.L.: A variational eigenvalue solver on a photonic quantum processor. Nature Commun. **5**, 1–7 (2014)
37. J. Gambetta, I. Faro and K. Wehden, "IBM's roadmap for building an open quantum software ecosystem," *IBM,* 4 (2021)
38. D. Gunlycke, M. C. Palenik, A. R. Emmert and S. A. Fischer, "Efficient algorithm for generating Pauli coordinates for an arbitrary linear operator," *arXiv preprint* arXiv:2011.08942*,* (2020)
39. C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio and P. J. Coles, "Variational quantum linear solver," *arXiv preprint* arXiv:1909.05820*,* (2019)
40. Andrea, "Pennylane," 20 January 2021. [Online]. Available: https://pennylane.ai/qml/demos/tutorial_vqls.html.

41. "IBM Quantum," 2021. [Online]. Available: https://quantum-computing.ibm.com/.
42. Cerezo, M., Sone, A., Volkoff, T., Cincio, L., Coles, P.J.: Cost function dependent barren plateaus in shallow parametrized quantum circuits. Nat. Commun. **12**, 1–12 (2021)
43. Sim, S., Johnson, P.D., Aspuru-Guzik, A.: Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. Adv. Quantum Technol. **2**, 1900070 (2019)
44. Spall, J.C.: An overview of the simultaneous perturbation method for efficient optimization. J. Hopkins APL Tech. Dig. **19**, 482–492 (1998)
45. Spall, J.C.: Implementation of the simultaneous perturbation algorithm for stochastic optimization. IEEE Trans. Aerosp. Electron. Syst. **34**, 817–823 (1998)
46. Cross, A.W., Bishop, L.S., Sheldon, S., Nation, P.D., Gambetta, J.M.: Validating quantum computers using randomized model circuits. Phys. Rev. A **100**, 032328 (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.