Key Challenges & Prospects for Quantum Computational Fluid Dynamics

René Steijl
James Watt School of Engineering
University of Glasgow
James Watt South Building
G12 8QQ Glasgow
United Kingdom

rene.steijl@glasgow.ac.uk

ABSTRACT

The development of quantum computing applications to Computational Fluid Dynamics has recently become an active area of research. Despite significant progress made so far, and the exciting prospects this approach offers, it is clear that a number of key challenges exist in the quest to derive quantum Computational Fluid Dynamics method with a significant (ideally exponential) complexity improvement over classical algorithms. The present set of notes provides a brief introduction to motivation and status of current work in this area, including a summary of the main challenges and prospects. Most of the current quantum computing applications to fluid dynamics involve hybrid quantum/classical algorithms. As an example of such an approach, the second part of the notes details early work by the author on a hybrid quantum/classical algorithm for the incompressible-flow Navier-Stokes equations. A Quantum Fourier Transform-based solution method for the Poisson equations occurring in the Vortex-In-Cell method is used, coupled to a classical implementation of the vortex-particle convection and viscosity terms. Examples for an application involving colliding vortex rings highlight the impact of 'quantum measurement' noise on the solution process. A short summary and outlook completes this set of notes.

Contents

1.0 INTRODUCTION

Quantum computing (QC)[1] is a rapidly developing area of research investigating computing hardware and algorithms that take advantage of quantum mechanical phenomena to perform computations. In recent years, significant progress has been made with building quantum computers. For a small number of applications, quantum algorithms have been developed that display a significant speed-up relative to classical methods. Computational quantum chemistry and quantum machine learning are among the areas of application receiving much recent research activity. Important developments for a wider range of applications include quantum algorithms for linear systems[2] and the Poisson equation[3]. Applications to computational science and engineering problems beyond quantum chemistry have only recently begun to appear[4–10]. Despite this research effort, progress in defining suitable engineering applications for quantum computers has been limited.

1.1 Motivation for using Quantum Computers and Quantum Computing

Traditionally, Computational Fluid Dynamics(CFD) applications have been among the most computationally demanding computer simulations in science and engineering. For the largest (Tier 0 and Tier 1) supercomputers, Direct Numerical Simulations (DNS) of turbulence, CFD modelling of chemically reacting flows and multi-phase flows typically constitute a significant part of the applications. Another trend can be identified: the longest (in terms of wall clock time) simulations routinely performed by academic researchers take on the order of weeks, and the arrival of faster supercomputers doesn't change this. Instead, the computational problems tackled increase with increasing computing power. In summary it can be said that for emerging supercomputer hardware, Computational Fluid Dynamics applications have always been an important driver. **For the emerging Quantum Computers, this does not yet seem to be the case.**

Beyond 'traditional' CFD applications, the modelling of hypersonic and rarefied flows using the Boltzmann kinetic equations is an example of an application where available computer resources are easily exhausted. Here, the challenge does not arise from large mesh sizes (i.e. billions of mesh points/cells in DNS), but from the combination of discretization of the computational domain as well as the phase (or, velocity) space. With the time dimension included, solution of the kinetic Boltzmann equation involves 7 dimensions. In a following set of lecture notes, Quantum Algorithms for Boltzmann kinetic equations are presented.

1.2 Background and context

Most current CFD solvers are based on the Finite-Volume Method. The time-integration method employed is a key factor in determining what forms the most CPU-time demanding part of a simulation:

- Using explicit time stepping, evaluation of numerical fluxes is typically the most CPU-time intensive part of the solver;

- Using implicit time stepping, the construction and solution of large (sparse) system of linear equations takes up the majority of CPU time;

Also, experience has shown that efficient use of emerging High Performance Computing platforms has been a key factor in CFD method development: parallel processing, as well as, the introduction of GPUs and FPGAs. From this previous CFD development, it follows that:

- Algorithms needed re-thinking, but overall Finite-Volume Method-based approaches with explicit/implicit time integration 'survived' in modified versions to enable implementation on the latest parallel computing hardware;

- Often, speeding-up linear system solution method on novel hardware proved a bigger challenge than gradient/flux calculations

This leads to an important current research question: **What will need to happen in terms of algorithm development for emerging quantum hardware?**

Using current and emerging Quantum Computing models, there are limited options when trying to stick to current CFD approaches:

- Using 'explicit' time stepping - the encoding of flux evaluations on quantum hardware is not promising. As an example, in a typical Finite Volume method (with 'cell-centred' or 'cell-vertex' based data), we work out fluxes using floating point additions and multiplications (non-linearity!). **Question: where is speed-up going to come from?**;

- Using 'implicit' time stepping - quantum linear system solvers (e.g. HHL) could be employed in coupled quantum/classical approaches. However, this approach has many challenges, as will be detailed later.

This leads to a further important current research question: **How can Quantum Computing benefit CFD using a more fundamental re-think of algorithms and methods? Also, what will need to happen for emerging quantum hardware to facilitate CFD quantum speed-up?**

1.3 Brief review of past work on Quantum Computational Fluid Dynamics

Early work in quantum computing relevant to the field of Computational Fluid Dynamics (CFD) mainly involved the work on quantum lattice-gas models by Yepez and co-workers[11, 12]. This work typically used type-II quantum computers, consisting of a large lattice of small quantum computers interconnected in nearest neighbour fashion by classical communication channels. In contrast to these quantum lattice-gas based approaches, most current research focuses on quantum algorithms designed for near-future 'universal' quantum computers. Previous work by the author introduced hybrid quantum/classical algorithms for fluid simulations based on quantum-Poisson solvers[7] as well as quantum algorithms for kinetic modelling of gases based on quantum walks[8, 9]. Recently, the quantum-circuit implementation of non-linear terms was considered[13]. An example of an application to a nonlinear fluid dynamics problem is the work of Gaitan[10], where for a very specific one-dimensional problem the complexity analysis showed potential for quantum speed-up. Recently, Carleman linearization techniques were used by Succi and co-workers[14] as a first step toward quantum algorithms for the Lattice Boltzmann Model (LBM) for (nearly) incompressible flows. Based on this short review, it is clear that a growing research community in Quantum Computational Fluid Dynamics can be identified.

1.4 Summary of promise and prospects offered by Quantum Computing

Quantum computing hardware and quantum algorithms are designed to take advantage of quantum mechanical phenomena to perform computations. For Quantum Computational Fluid Dynamics, the main quantum mechanical effects used are:

- Quantum Parallelism: a coherent quantum state in a multi-qubit register with nq qubits theoretically offers 2^{nq} degrees-of-freedom to use in the simulation - this **exponential growth with number of qubits** is a key factor in achieving quantum speed-up;
- Quantum entanglement: a unique feature of quantum mechanics, not available in 'classical' computing hardware

Other effects, such as quantum teleportation, have so far not found a major role in quantum algorithms performing computational engineering computations.

1.5 Summary of key challenges

With the exception of quantum measurement operations, quantum mechanical operators are unitary, linear and reversible. A further complicating factor is the **'no-cloning theorem'**. To illustrate how this is a challenge for implementing CFD methods, consider the following example. In most CFD solvers, subroutines or functions exist that take a variable, e.g. u, then create a temporary copy $temp = u$, to perform the computation of the value of u^2 as $u \times temp$. Clearly, the no-cloning theorem directly prevents a similar action to be encoded in Quantum Computing.

Also, for data encoded in terms of the complex amplitudes of the Schrödinger wavefunction, there is a need for this state vector to have a unit norm, since these amplitudes represent probabilities of states. This means that for an operator attempting to compute the squares of these complex amplitudes, the resulting state vector loses its unit norm. So, even without the no-cloning theorem complicating such a step, this points to a further problem with computing non-linear terms. This loss of unit norm of the quantum state vector represents an example of a **non-unitary operation**.

So, for Quantum Computational Fluid Dynamics, **dealing with non-linear terms inevitable in fluid mechanics modelling, can be regarded as one of the main challenges**.

For most of the work in Quantum Computational Fluid Dynamics, a **hybrid quantum/classical approach** is used, with the quantum processor performing computations for which efficient quantum algorithms exist, while the output is then passed on to classical hardware to perform further computational tasks. This typically produces new 'classical' data that then needs to be used to re-initialize the quantum state in the quantum processor. To make this challenge particularly significant, most hybrid quantum/classical approaches employ a repeated exchange of quantum-to-classical and classical-to-quantum information. In summary, key challenges for the **hybrid quantum/classical approach** are:

- Cost and complexity of (repeated) measurements;
- Statistical noise created by sampling the quantum solution;
- Cost and complexity of (repeated) re-initialization.

As an illustration of the problems created by statistical noise originating from the sampling of the quantum solution, the final part of these notes discusses an example hybrid quantum/classical approach for the Navier-Stokes equations employing a QFT-based Poisson solver.

2.0 'HIGH-LEVEL' QUANTUM-CIRCUIT IMPLEMENTATION OF QUANTUM ALGORITHMS

In Quantum Computational Fluid Dynamics, it can be expected that the Quantum Circuit model will continue to play an important role. One of the useful features of this model is the potential to use of 'high-level' circuit programming along with circuit transformation tools (sometimes referred to as 'transpilers'). This enables a quantum algorithm developer to create quantum circuits including complex multi-qubit quantum gate operations that are not implemented as 'native' gates on the target quantum hardware. This allows the programmer to use a higher level of abstraction than when creating quantum-circuit implementations in terms of the available native gates. The analogy with high-level programming languages is clear - CFD algorithms are typically written in Fortran, C or C++, and efficient compilers then create the executable code for the target computer hardware. It can be expected that the programming 'tool chain' for quantum algorithm development will continue to develop quickly and involve different levels of abstraction.

2.1 Example: 'streaming' operation on 1D periodic domain

As a first illustration of this 'high-level' quantum-circuit implementation approach we consider a quantum circuit performing 'streaming' operations on a periodic one-dimensional domain with 64 lattice sites. The 'left' streaming moves a value from its original lattice site to its left-nearest neighbour, while the 'right' streaming performs such a move to the right-nearest neighbour. The quantum circuits shown in Figure 1 uses the quantum amplitude-based encoding approach, so that the coherent 6-qubit register has 64 degrees-of-freedom - sufficient for a scalar value in each of the 64 lattice sites. A key feature of the quantum circuits is the occurrence of multiply-controlled Toffoli gates. By adding 'ancillae' qubits, the quantum-circuit implementation can be transformed to reduce the gate operations to fewer control qubits. This is illustrated in Figure 2.

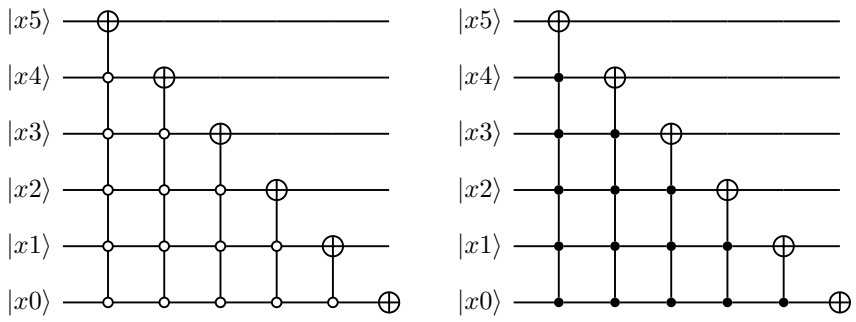


Figure 1: Streaming of scalar data on uniformly-space 1D mesh (64 cells encoded with 6 qubits) with periodic boundary conditions. Circuits for 'Left' and 'Right' streaming. Most-significant qubit as used in data ordering is at the top of circuits.

2.2 Example: multiply-controlled Toffoli gate

The previous section illustrated how quantum circuits with Toffoli gates with up to 5 control qubits can be transformed to limit the number of control qubits to ≤ 3 by addition of 3 ancillae qubits. Clearly, even after this transformation the 3-qubit controlled Toffoli gates are still far removed from the gate operations implemented in representative quantum hardware. Also, we need to consider the effect of quantum errors (de-coherence, etc.) on the gate operations taking place in the quantum hardware. Therefore, fault-tolerance is a key factor in

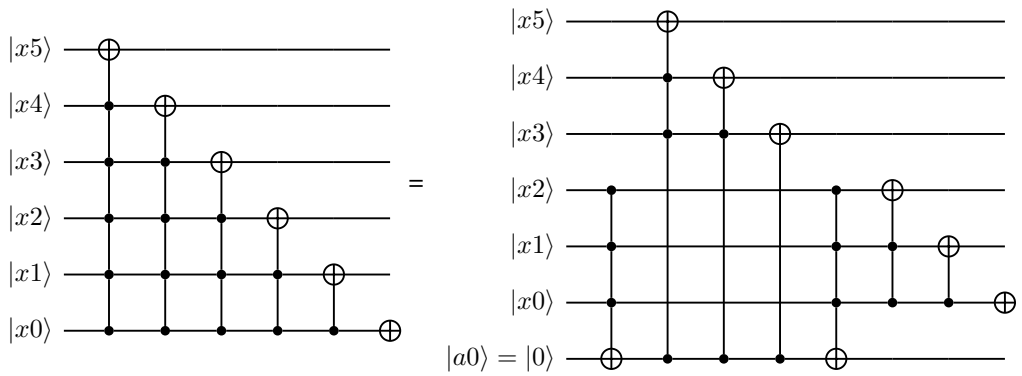


Figure 2: Streaming of scalar data on uniformly-space 1D mesh (64 cells encoded with 6 qubits) with periodic boundary conditions. Circuit for 'Right' streaming was transformed to reduce number of control qubits to ≤ 3. Most-significant qubit as used in data ordering is at the top of circuits.

hardware implementations of quantum gates. Recently, Biswal et al.[15] presented techniques to create fault-tolerant implementations of multi-controlled Toffoli gates. For the Toffoli-4 gate, i.e. the Toffoli gate with 3 control qubits, Figure 3 and Figure 4 show example decompositions facilitating fault-tolerant implementations involving representative 'native' gates.

These examples show that as long as the transformation or 'transpilation' to such fault-tolerant native gates can be hidden from the quantum algorithm developer, it makes sense for quantum algorithm researchers using the Quantum Circuit Model to use this 'high-level abstraction' approach.

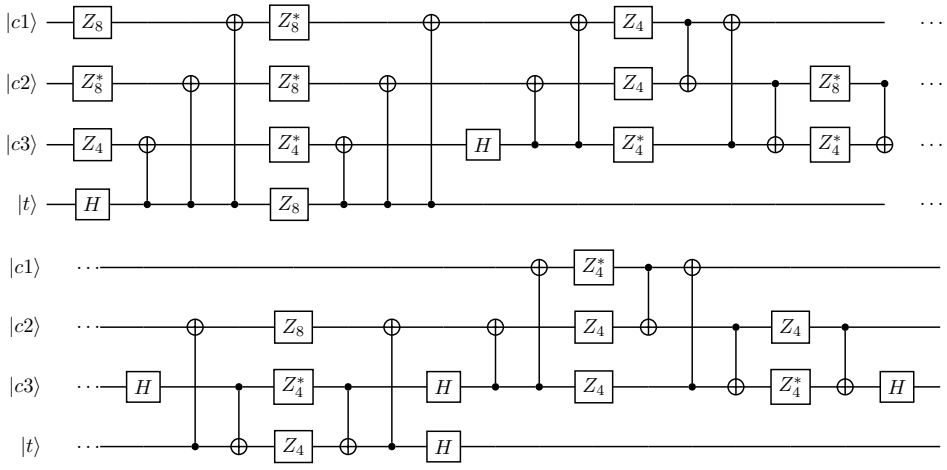


Figure 3: Decomposition of Toff4 - Biswal (2019) - Fig.5

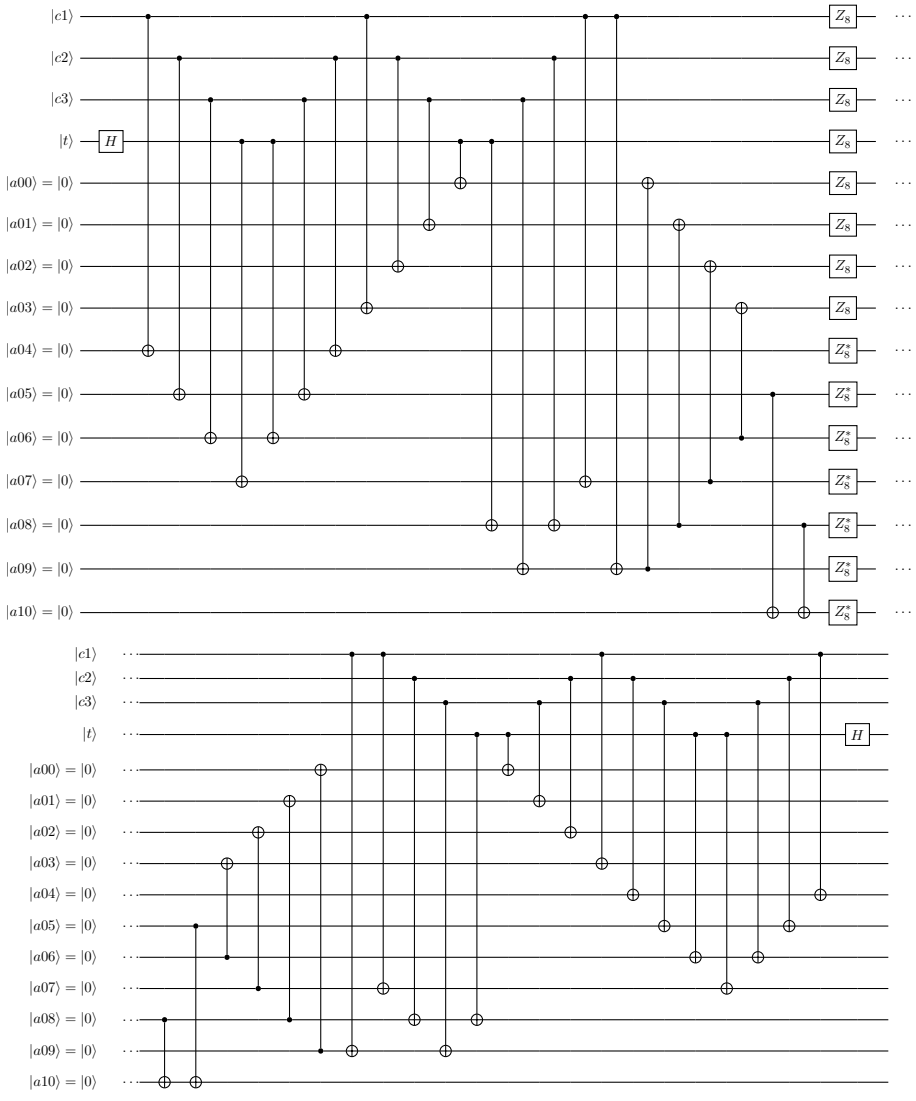


Figure 4: Decomposition of Toff4 - Biswal (2019) - Fig.10

3.0 VORTEX-IN-CELL METHOD WITH QUANTUM POISSON SOLVER

The vortex-in-cell method[16, 17] is a well-studied hybrid particle-mesh method for incompressible flows and is particularly well-suited for flows in regular domains such that efficient Poisson solvers can be used. In the present work, the Fourier Analysis approach to solving the problem in a fully periodic domain is used, using the Quantum Fourier Transform (QFT) for the required discrete Fourier Transforms. The hybrid quantum/classical solver was implemented in C++, where the quantum-circuits implementing the QFT and its inverse were simulated using a Quantum Computer Simulator built-in the C++ CFD framework[7].

The vortex-in-cell (VIC) method solves the incompressible-flow Navier-Stokes equations transformed into the Helmholtz equation for vorticity evolution,

$$\frac{\partial \omega}{\partial t} + \big(\mathbf{u}\cdot\nabla\big)\omega = \big(\omega\cdot\nabla\big)\mathbf{u} + \nu\Delta\omega \quad ; \quad \omega = \nabla\times\mathbf{u} \tag{1}$$

A viscous splitting algorithm is employed, which first solves the inviscid equation,

$$\frac{\partial \omega}{\partial t} + \big(\mathbf{u}\cdot\nabla\big)\omega = \big(\omega\cdot\nabla\big)\mathbf{u} \tag{2}$$

where the right-hand side represents the vortex-line stretching term. Next, the viscous effects are introduced by solving the diffusion equation,

$$\frac{\partial \omega_i}{\partial t} = \nu\Delta\omega_i \tag{3}$$

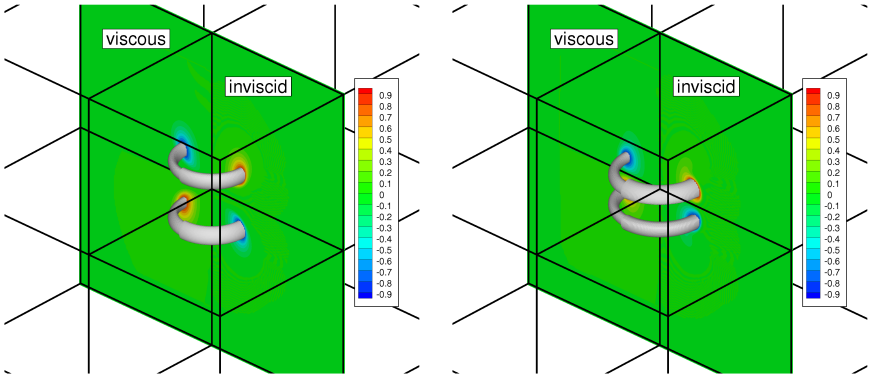
where ω_i is the vorticity field obtained after the inviscid step. The advection of vorticity represented by the second term on the left-hand side of Equation (2) is modelled using Lagrangian motion of vortex particles on a regular background mesh. The vortex-line stretching term is evaluated on this regular mesh at the start of the time-step. In the current implementation, the vorticity is transferred from the particle to the mesh at the end of each time-step. Since this is done after each step of the inviscid equation, the viscous effects can be introduced by Equation (3), discretized on the regular background mesh using second-order accurate central differences. This is followed by a regularization step in which the particles are placed at the nodes of the regular background mesh. The vorticity represented on this mesh is then used to re-initialize the strength of the vorticity in each of the vortex particles. The velocity field is then evaluated from a vector potential $\mathbf{A}$, which follows from solving the following three Poisson problems,

$$\begin{aligned} \Delta A_x &= -\omega_x \\ \Delta A_y &= -\omega_y \\ \Delta A_z &= -\omega_z \\ \mathbf{u} &= \nabla\times\mathbf{A} \end{aligned} \tag{4}$$

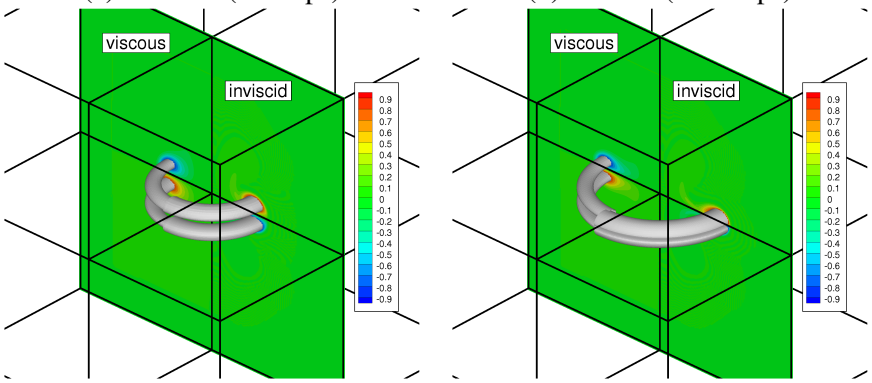
The solution of the Poisson problems forms the quantum part of the present hybrid quantum/classical algorithm. For the time-integration, a predictor-correction approach is used, such that for each time step, the steps described above need to be executed twice. For the transfer of vorticity from particles-to-mesh and mesh-to-particles, the following interpolation kernel (M_4') is used,

$$\zeta(r/\sigma) = \begin{cases} 1 - \frac{5}{2}\big(r/\sigma\big)^2 + \frac{3}{2}\big(r/\sigma\big)^3 & \text{if } 0 \le r/\sigma \le 1 \\ \Big[2 - \big(r/\sigma\big)^2\Big]^2\Big(1 - \big(r/\sigma\big)\Big)/2 & \text{if } 1 \le r/\sigma \le 2 \\ 0 & \text{if } 2 \le r/\sigma \end{cases} \tag{5}$$

where r is the distance from the center of the kernel and σ is the mesh width. For the present 3D cases, a tensor-product of this kernel is used.



(a) $T = 6.0$ (60 steps) (b) $T = 12$ (120 steps)



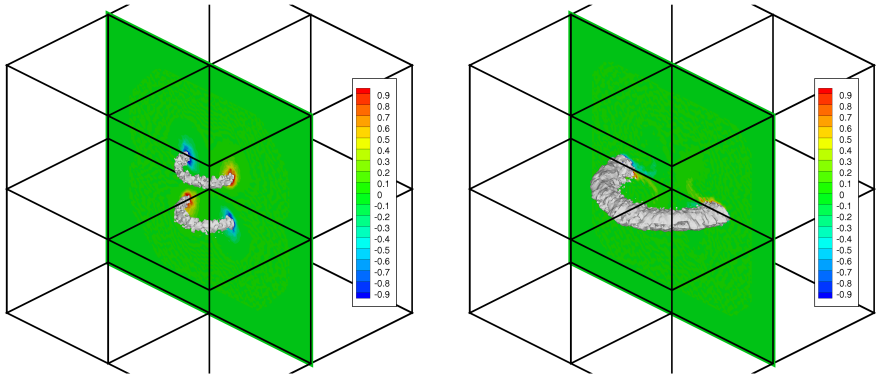
(c) $T = 18$ (180 steps) (d) $T = 24$ (240 steps)

Figure 5: Noise-free simulation of colliding vortex rings simulated with and without viscosity. Vorticity isosurface at $\omega = 1$. Uniform 128^3 background mesh was used.

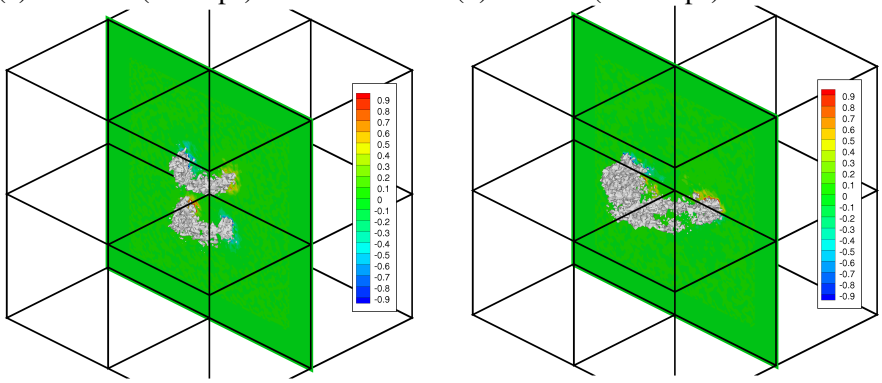
3.1 Quantum measurement of solution of Poisson equations - effect of noise

In the hybrid quantum/classical VIC-based Navier-Stokes solver, **data exchange** between the quantum processor(s) and the classical computation can be summarized as follows:

- 'Classical' data is used as source terms for the 3 Poisson problems. Specifically, the vorticity vector ω evaluated after vortex-particle convection and viscous effects have been accounted for, needs to be used in re-**initialization** of the QFT-based Poisson solver. **This step is expensive due to the overhead involved in initialization of the quantum register(s), but should not introduce noise**;
- The solution to the 3 Poisson problems is amplitude-encoded in the quantum state vector representing $\mathbf{A}$;
- After completing the Poisson solutions, velocity field is to be evaluated from: $\mathbf{u} = \nabla \times \mathbf{A}$;
- $\mathbf{u}$ is needed by 'classical' parts of the algorithm. **Measurement of the quantum state vector will create a 'noisy' solution. Also, need for statistical sampling creates significant computational overhead - need for multiple realizations**.

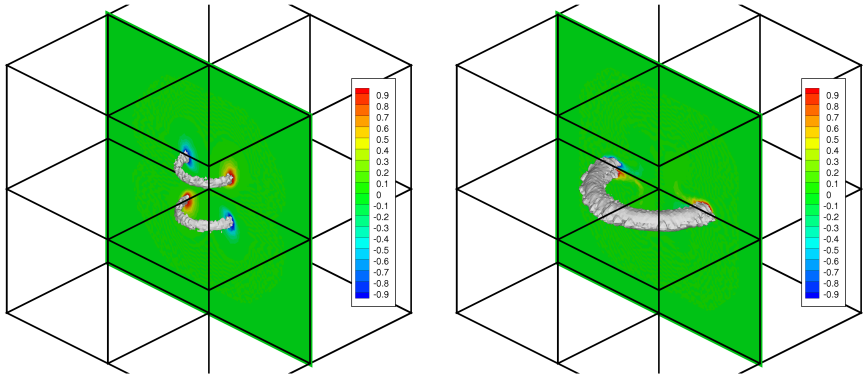


(a) $T = 6.0$ (60 steps) - noise 0.01% (b) $T = 24$ (240 steps) - noise 0.01%



(c) $T = 6.0$ (60 steps) - noise 0.05% (d) $T = 24$ (240 steps) - noise 0.05%

Figure 6: Colliding vortex rings with simulated noise. (a)-(b) scaled noise magnitude 0.0001, (c)-(d) scaled noise magnitude 0.0005. Vorticity isosurface at $\omega = 1$. Uniform 128^3 background mesh was used. Inviscid flow simulations.



(a) $T = 6.0$ (60 steps) - noise 0.1% (b) $T = 24$ (240 steps) - noise 0.1%

Figure 7: Colliding vortex rings with simulated noise. Velocity field evaluated in spectral space. Scaled noise magnitude 0.001. Vorticity isosurface at $\omega = 1$. Uniform 128^3 background mesh was used. Inviscid flow simulations.

Clearly, the evaluation $\mathbf{u} = \nabla \times \mathbf{A}$, with $\mathbf{A}$ encoded in terms of amplitudes of the quantum state vector is a key part of the design of the hybrid quantum/classical VIC method. In the work previously published[7], **two different versions** were evaluated:

- **Version 1**: In the first version of the vortex-in-cell method considered here, the velocity field is obtained from the vector potential as shown in Equation (4), where the gradients of the potential are evaluated on the Cartesian mesh using second-order accurate finite-differences. **Key problem: the finite-difference computation of the velocity is performed on the classical computer, so that noise levels are amplified relative to the levels in the original vector potential as a result of applying finite-differences to noisy data**;
- **Version 2**: Improved design aiming to eliminate the use of finite-differences on the vector potential. Since the Poisson solver used in the considered rectangular Cartesian domains is based on the discrete Fourier transform, here implemented using the QFT, the second version of the vortex-in-cell method was designed to evaluate the velocity field from the Fourier coefficients of the solution to the Poisson equation of Equation (4). **Key challenge: can gradient computation (non-unitary operation) be performed efficiently in Fourier-space on the quantum processor?**

In the second design, the derivatives of the vector potential in one of the three coordinate directions are obtained from the coefficients of the Discrete Fourier Transform has been applied using the QFT. Then, as a first step, the following single-qubit operator is applied to the first qubit in the relevant range of qubits encoding the considered coordinate direction:

$$\begin{pmatrix} -i & 0 \\ 0 & i \end{pmatrix} \tag{6}$$

This step can be efficiently executed on a quantum computer, since it only involves a single-qubit operation. However, this is only the first step. In the second step, the quantum Fourier components encoded as quantum state vector amplitudes need to be multiplied by their respective wave number. The derivatives then follow from performing an inverse QFT. **A key challenge here is that the multiplication-by-wavenumber operation is non-unitary**.

3.2 Example test case: colliding vortex rings

A pair of colliding vortex rings is considered with equal but opposite-signed strength. A non-dimensional formulation is used with the flow domain a unit cube and the ratio of vortex core radius (r_c) to ring radius (R) equal to $r_c/R = 1/5$. For a uniform 128^3 mesh the vortex core radius is 5 mesh widths, while the vortex core radius is 10 mesh widths for a uniform 256^3 mesh. The vertical spacing between the centers of the vortex rings at the start of the simulation is 1.5 times the ring radius. The vortex rings are initialized with a Gaussian profile with a unit (non-dimensional) peak vorticity. Inviscid results and results for a circulation based Reynolds number of 10^5 are compared in Figure 5, showing the increasing radius for both rings due to mutual velocity induction. For the inviscid result the vorticity magnitude increase due to vortex-line stretching effect is somewhat more pronounced than the viscous simulation where part of the increase is cancelled by viscous dissipation. **Note: In Figure 5, the solution of the Poisson systems was assumed noise-free, so that Version 1 and Version 2 of the hybrid quantum/classical VIC solvers give the same result**. Next, the sensitivity to operational errors of the quantum gates and statistical sampling errors due to measurement is investigated. The vector potential is computed from the solution of a Poisson with the vorticity forming the source term as in Equation 4. Since ideal quantum gate operations involve unitary operations, the algorithm used here first normalizes the system and stores the scaling factor involved. Then random noise is added to the solution of the Poisson system before the scaling factor is applied in the reverse direction. Version 1 of the vortex-in-cell method then applies second-order finite differences to create an updated velocity field defined

on the background mesh, which is subsequently transferred to the vortex particles for the advection step in the next time step. Here two amplitudes of the random noise were considered, e.g. 0.0001 and 0.0005 relative to the normalized state vector in the quantum network representing the QFT based Poisson solver. Figures 6 show the results for both noise amplitudes at two non-dimensional time instances for Version 1 of the hybrid quantum/classical solver. Clearly, the larger noise amplitude considered leads to significant errors early in the simulation and at later stages both vortex rings have lost their coherence. In contrast, the smaller noise amplitude enables the simulation to maintain the coherence of the vortex rings as well as the main fluid dynamics aspects of the interaction. For Version 2, a similar study of the sensitivity to measurement-induced statistical noise was performed. Figure 7 shows results for this version, where the noise level is increased to 0.001. The results show a coherence of the computed vortex system similar to that for the lowest noise level, i.e. 0.0001, for Version 1 of the solver. These results clearly show that with suitable modifications to this particular vortex-in-cell implementation the sensitivity to noise and uncertainty can effectively be reduced.

3.3 Key findings from hybrid quantum/classical VIC Navier-Stokes solver

In summary, the second implementation using Fourier-space based evaluation of $\mathbf{u}$ compares to the version using a finite-difference approach as follows:

- the finite-difference method applies finite-difference operations on a 'noisy' set of data (representing the vector potential) defined on the three-dimensional mesh. This process will typically amplify the noise levels of the derivative as compared to the original signal;
- the QFT-based approach for derivatives avoids this noise-amplifying step;
- the single-qubit operation required for the derivative-calculation can be done efficiently on a quantum computer;
- the main drawback of the QFT-based approach as implemented here, is that it assumes that the multiplication with the wave numbers can be done efficiently on the classical hardware in the envisaged hybrid approach. Furthermore, it has so far been assumed that no additional noise is created in the transfer of the Fourier coefficients from quantum to classical hardware.

It follows that the proposed QFT-based approach for the velocity-field evaluation would mainly be of interest for quantum computer applications in case the multiplication with the wave numbers can be performed completely on a quantum computer.

4.0 SUMMARY AND OUTLOOK

In the present set of notes, the main motivation for research into Quantum Computational Fluid Dynamics was outlined. The key features of quantum computing offering the potential for significant quantum speed-up were presented. Crucially, a significant number of key challenges remain to resolved before Quantum Computational Fluid Dynamics applications can be realized with meaningful 'quantum supremacy'.

The example hybrid quantum/classical Vortex-In-Cell Navier-Stokes solver presented here, highlighted some of the key challenges associated with this hybrid approach. **The computational overhead of repeated re-initializations of the quantum system as well as the need for statistical sampling of multiple realizations form the main problems with hybrid quantum/classical approaches**.

In a further set of lecture notes in the current Lecture Series, initial research efforts by the author and co-workers is presented aiming to create **quantum algorithms that minimize the quantum/classical data exchange**. Clearly this approaches poses many challenges as well. Although some advances have recently been made for lattice-based modelling approaches, it is clear that further investigations are needed to determine the potential of this approach.

ACKNOWLEDGMENTS

The author would like to acknowledge the financial support provided by the University of Glasgow.

REFERENCES

[1] Nielsen, M. and Chuang, I., *Quantum Computation and Quantum Information: 10th Anniversary Edition*, Cambridge University Press, 2010.

[2] Harrow, A., Hassidim, A., and Lloyd, S., "Quantum Algorithm for Linear Systems of Equations," *Phys. Rev. Lett.*, Vol. 103, 2009, pp. 150502.

[3] Cao, Y., Papageorgiou, A., Petras, I., Traub, J., and Kais, S., "Quantum Algorithm and Circuit Design solving the Poisson Equation," *New Journal of Physics*, Vol. 15, 2013, pp. 013021.

[4] Montanaro, A. and Pallister, S., "Quantum algorithms and the finite element method," *Phys. Rev. A*, Vol. 93, Mar 2016, pp. 032324.

[5] Scherer, A., Valiron, B., Mau, S., Alexander, S., van den Berg, E., and Chapuran, T., "Concrete resource analysis of the quantum linear-system algorithm used to compute the electromagnetic scattering cross section of a 2D target," *Quantum Information Processing*, Vol. 16, No. 3, Jan 2017, pp. 60.

[6] Xu, G., Daley, A., Givi, P., and Somma, R., "Turbulent Mixing Simulation via a Quantum Algorithm," *AIAA Journal*, Vol. 56, No. 2, 2018, pp. 687–699.

[7] Steijl, R. and Barakos, G., "Parallel evaluation of quantum algorithms for computational fluid dynamics," *Computers & Fluids*, Vol. 173, 2018, pp. 22–28.

[8] Todorova, B. and Steijl, R., "Quantum Algorithm for the collisionless Boltzmann equation," *J. Comp. Phys.*, Vol. 409, 2020, pp. 109347.

[9] Steijl, R., "Quantum Algorithms for Fluid Simulations. In: Bulnes, F. (ed.) Advances in Quantum Communication and Information. IntechOpen. ISBN 9781789852684 (doi: 10.5772/intechopen.86685," 2020.

[10] Gaitan, F., "Finding flows of a Navier–Stokes fluid through quantum computing," *npj Quantum Information*, Vol. 6, 2020, pp. 61.

[11] Yepez, J., "Quantum lattice-gas model for computational fluid dynamics," *Phys. Rev. E*, Vol. 63, 2001, pp. 046702.

[12] Berman, G., Ezhov, A., Kamenev, D., and Yepez, J., "Simulation of the diffusion equation on a type-II quantum computer," *Phys. Rev. A*, Vol. 66, 2002, pp. 012310.

[13] Steijl, R., "Quantum algorithms for nonlinear equations in fluid mechanics. In: Zhao, Y. (ed.) Quantum Computing and Communications. IntechOpen:London. ISBN 9781839681332 (doi: 10.5772/intechopen.95023)," 2022.

[14] Itani, W. and Succi, S., "Analysis of Carleman Linearization of Lattice Boltzmann," *Fluids*, Vol. 7, No. 1, 2022.

[15] Biswal, L., Bhattacharjee, D., Chattopadhyay, A., and Rahaman, H., "Techniques for fault-tolerant decomposition of a multicontrolled Toffoli gate," *Phys. Rev. A*, Vol. 100, Dec 2019, pp. 062326.

[16] Cottet, G. and Poncet, P., "Simulation and Control of Three-Dimensional Wakes," *Computers and Fluids*, Vol. 33, No. 5, 2004, pp. 697–713.

[17] Kosior, A. and Kudela, H., "The 3D Vortex Particle Method in Parallel Computations on Many GPUs," *Computers and Fluids*, Vol. 92, 2014, pp. 274–280.