QUANTUM IN COMPUTATIONAL FLUID DYNAMICS

Introduction:

What is CFD?

Computational Fluid Dynamics is a branch of fluid mechanics that uses numerical analysis and data structures to analyze and solve problems involving fluid flow. The fundamental basis of all CFD problems is the Navier-Stokes equations which define many single-phase which may define many single phase fluid flows. This technique is powerful and spans a wide range of industrial and non-industrial application areas. Examples of such applications are-

- Aerodynamics of aircraft and vehicles: lift and drag
- Hydrodynamics of ships
- Power plant: combustion in internal combustion engines and gas turbines
- Turbo machinery: flows inside rotating passages, diffusers etc.
- Electrical and electronic engineering: cooling of equipment including microcircuits
- Chemical process engineering: mixing and separation, polymer moulding.
- External and internal environment of buildings: wind loading and heating/ventilation.
- Marine engineering: loads on off-shore structures
- Environmental engineering: distribution of pollutants and effluents
- Hydrology and oceanography: flows in rivers, estuaries, oceans
- Meteorology: weather prediction
- Biomedical engineering: blood flows through arteries and veins

The need for CFD-

Applying the fundamental laws of mechanics to a fluid gives the governing equations for a fluid. The conservation of mass equation is-

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{V}) = 0$$

and the conservation of momentum equation is-

$$\rho \frac{\partial \vec{V}}{\partial t} + \rho(\vec{V} \cdot \nabla)\vec{V} = -\nabla p + \rho \vec{g} + \nabla \cdot \tau_{ij}$$

These equations along with the conservation of energy equation form a set of coupled, nonlinear partial differential equations. It is not possible to solve these equations analytically for most engineering problems. However, it is possible to obtain approximate computer-based solutions to the governing equations for a variety of engineering problems. This is the subject matter of Computational Fluid Dynamics (CFD).

Theory Regarding CFD-

A mathematical model for fluid flow is given by Navier-Stokes Equation.

The Navier–Stokes equations mathematically express conservation of momentum and conservation of mass for Newtonian fluids. They are sometimes accompanied by an equation of state relating pressure, temperature and density. They arise from applying Isaac Newton's second law to fluid motion, together with the assumption that the stress in the fluid is the sum of a diffusing viscous term (proportional to the gradient of velocity) and a pressure term—hence describing *viscous flow*.

These equations are defined as-

$$\rho\frac{D\mathbf{u}}{Dt} = \rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \nabla\cdot\left\{\mu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^{\mathrm{T}} - \tfrac{2}{3}(\nabla\cdot\mathbf{u})\mathbf{I}\right) + \zeta(\nabla\cdot\mathbf{u})\mathbf{I}\right\} + \rho\mathbf{g}.$$

Limitations of CFD-

- Errors may occur due to simple flow models or simplified boundary conditions.
- Possible uncertainties caused by too little computing values per cell and hence therefore resulting interpolation errors.
- Computation time may extend for large models.
- The costs may be much higher due to wrong consulting compared to experiments.

Challenges of CFD-

The computational simulation of fluid flow presents a number of severe challenges for algorithm design. At the level of inviscid modeling, the inherent nonlinearity of the fluid flow equations leads to the formation of singularities such as shock waves and contact discontinuities. Moreover, the geometric configurations of interest are extremely complex, and generally contain sharp edges which lead to the shedding of vortex sheets. Extreme gradients near stagnation points or win tips may also to numerical errors that can have global influence. Numerically generated entropy may be convected from the leading-edge, for eg: causing the formation of a numerical induced boundary layer which can lead to separation. The need to treat exterior domains of infinite extent is also a source of difficulty. Boundary conditions imposed at artificial outer boundaries may cause reflected waves which significantly interfere with the flow. When viscous effects are also included in the simulation, the extreme difference of the scales in the viscous boundary layer and the outer flow, which is essentially inviscid, is extreme variations in the mesh intervals. For these reasons, CFD has been a driving force for the development of numerical algorithms. Now out of all problems, quantum computing approach has been taken in some issues regarding improvement of the solvers which can be used, but still now other domains of problems in CFD remains unsolved.

What is Quantum Computing?

Quantum computing is the exploitation of collective properties of quantum states, such as superposition and entanglement, to perform computation. The devices that perform quantum computations are known as **quantum computers**. They are believed to be able to solve certain computational problems, such as integer factorization (which underlies RSA encryption), substantially faster than classical computers. The study of quantum computing is a subfield of quantum information science.

The fundamental unit of quantum computation is the quantum bit or qubit. Whereas a classical bit is confined to existing in either the 0 or 1 state, a qubit can be in a state of superposition, i.e. it exists in both states simultaneously. Upon measuring the qubit, the quantum state collapses to either of these two states, and the qubit is no longer in a state of superposition. The state of a qubit is defined through a pair of complex numbers c_0 and c_1 such that the probability of finding the qubit after measuring in state 0 is |c0|^2 and the probability of measuring state 1 is |c1|^2. For a quantum system with n coherent qubits, measurement of the quantum system can lead to 2^n possible outcomes.

Quantum Algorithm being used in CFD domain-

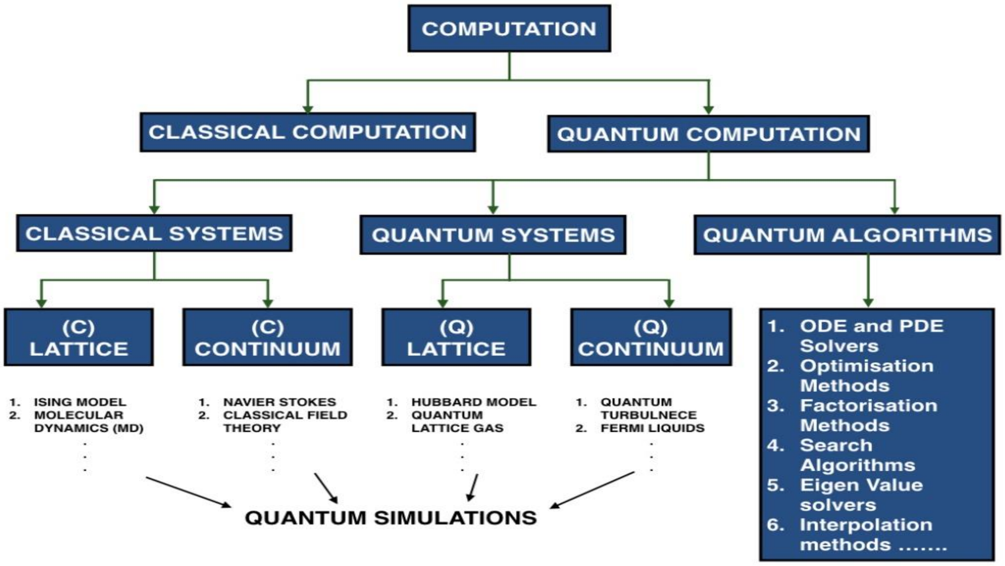
In CFD domain the systems can be divided into two systems-1) Classical System 2) Quantum System. Quantum implementation of both such systems are done but to a certain complexity level.

- **Classical Systems-** A slightly harder but an interesting avenue would be to compute classical systems using quantum computing. Most of the effort here would be spent in translating classical dynamics into the quantum language. One can then harness the quantum advantage from here on. Once again, a similar classification could be done, where one looks at lattice systems, Lattice Boltzmann Methods and molecular dynamics, or one can start using quantum based mathematical tools such as ODE solvers and eigenvalue solvers or optimization methods for integrating and solving classical governing equations such as the Navier-Stokes equations.
- **Quantum Systems**-Quantum systems are obvious candidates for using QC. Though all quantum systems are legitimate candidates, problems that are currently being explored, or could be explored, fall in two categories:

1) *Lattice based systems*: Most hard quantum condensed matter systems such as the Hubbard problem or the quantum lattice gas fall in this category. Here, one can look at the lattice based Hamiltonians to either perform a quantum simulation or compute observables and properties via specific algorithms.
2) *Continuum problems*: On the other hand, some problems such as quantum turbulence and quantum liquids, would require the integration of the many-body Schrodinger equation followed by a mapping to macroscopic observables. One could also use quantum algorithmic numerical tools to

integrate model equations such as Gross Pitaevski equations (in the case of quantum turbulence) or do a quantum Monte Carlo study, etc.

In this report, quantum computation methods for continuum flow classical systems are defined and implemented. Models based on Navier-Stokes Equation are implemented through Quantum Fourier Transform and Approximate Quantum Fourier Transform. Another model for Navier-Stokes at kinetic level was described by Discreet Velocity Method and its circuit was implemented inspired on the theory of Quantum Walk on Dirac Equation.



Classification of problems based on QC methods

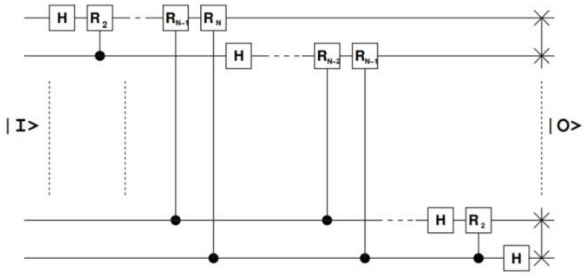
Methods used:

Quantum Fourier Transform:

The Quantum Fourier Transform (QFT) provides an efficient method for computing the Discrete Fourier Transform.

$$QFT(|\psi\rangle) = \sum_{j=0}^{N-1} b_j \, |j\rangle \qquad ; \qquad b_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} c_k \cdot \exp\left(\frac{2\pi i \cdot j \cdot k}{N}\right)$$

The Quantum circuit model can be represented as:



Circuit representation of the Quantum Fourier transform. The input quantum state $|I\rangle$ is transform into an output quantum state $|O\rangle$. In the output state vector represented by 2^{n_q} complex numbers, vector elements appear in bit-reversed order. This is represented by the crosses in the circuit output channels.

The circuit takes the circuit involves a series of gate operations, e.g. H and Rn, where H represents the single-qubit Hadamard operation and Rn the controlled-rotation acting on two qubits. Rn performs a conditional phase shift, i.e. it applies a phase factor exp(2πi/2^n) only if the considered two qubits are both in their |1> state, leaving the other three basis states unaffected. The single-qubit Hadamard transformation and two-qubit controlled rotation operations are defined as-

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad ; \quad R_n = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{2\pi i/2^n} \end{pmatrix} l$$

Quantum Algorithm in Vortex-in-cell Method

Hybrid quantum/classical approach in vortex-in-cell method

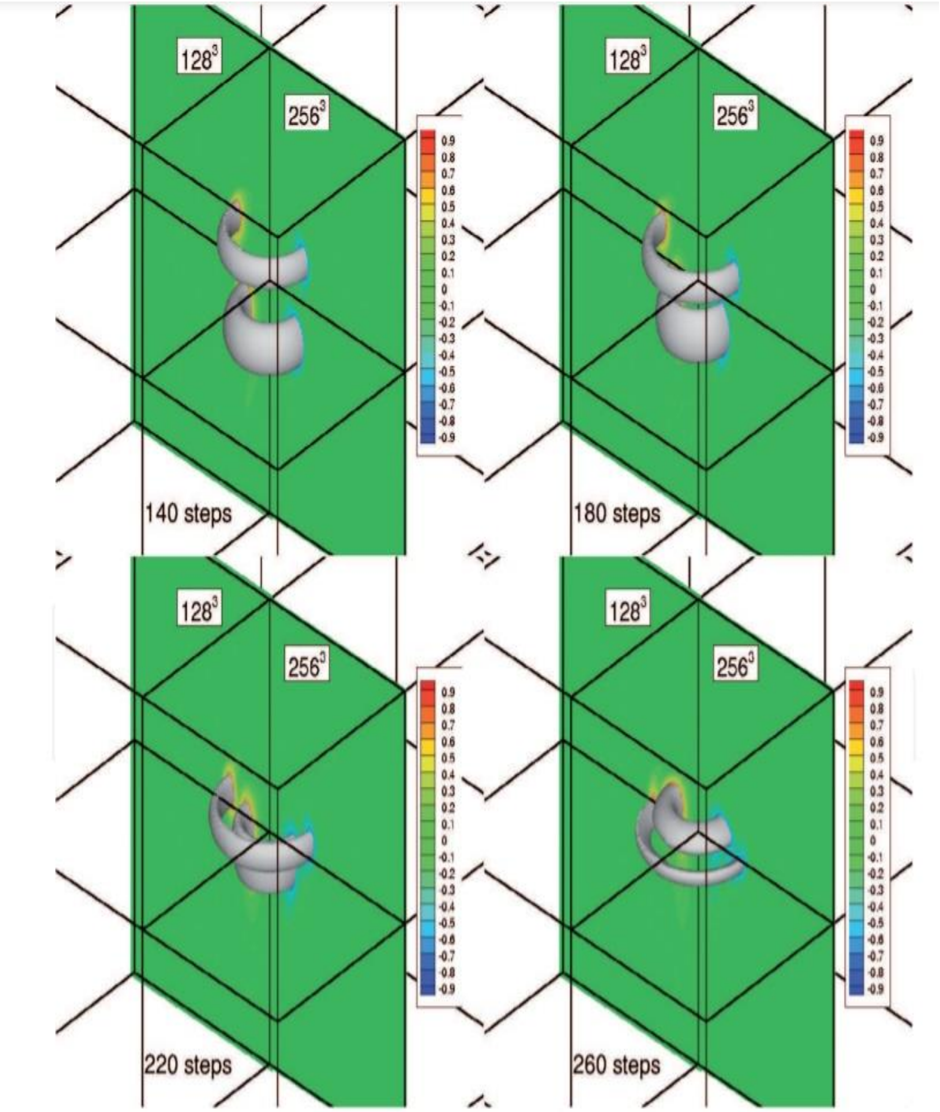
Vortex in cell method is a well studied hybrid particle mesh method to incompressible flows and is particularly well-suited for flows in regular domains such that efficient poison solvers can be used. We know that Fourier analysis approach is used for solving problem in fully periodic domain thus QFT is used for Discreet Fourier transforms. The vortex in the shell method solves the incompressible-flow Navier-Stokes equations transformed into Helmholtz Equations for vorticity evolution.

$$\frac{\partial\vec{\omega}}{\partial t} + \vec{u}\,\frac{\partial\vec{\omega}}{\partial\vec{x}} = (\vec{\omega}.\nabla)\vec{u} + \nu\Delta\vec{\omega}; \vec{\omega} = \nabla x\, \vec{u}$$

In simulations using the VIC, the flow evolves through a (large) number of time steps. During each of these time steps, the velocity field was recomputed using solutions of three Poisson problems for stream function A :

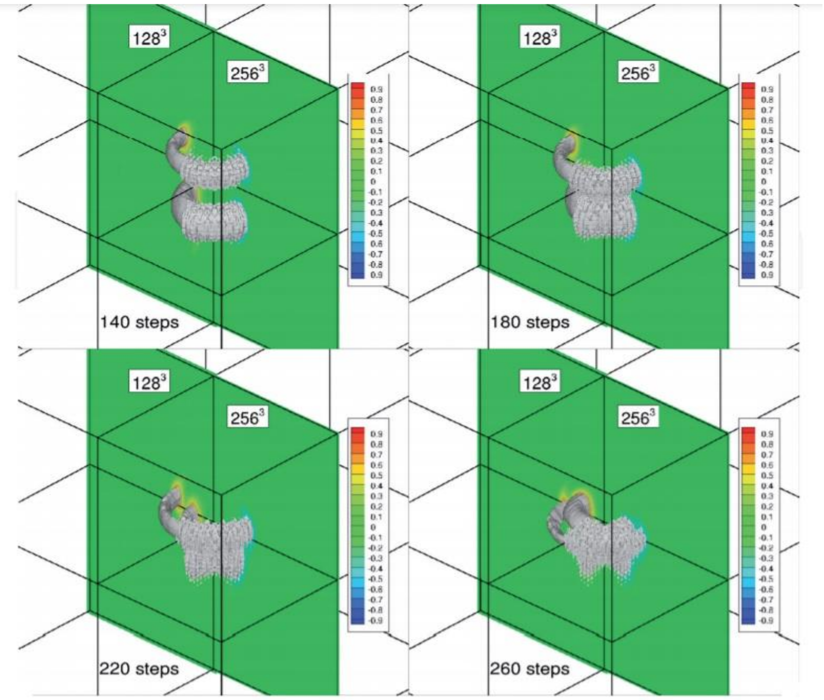
$$\Delta\vec{A} = -\vec{\omega}; \vec{u} = \nabla x\, \vec{A}$$

This part of the VIC is of particular interest here, as it represents the part that would be performed by the quantum processor in the quantum coprocessor model.



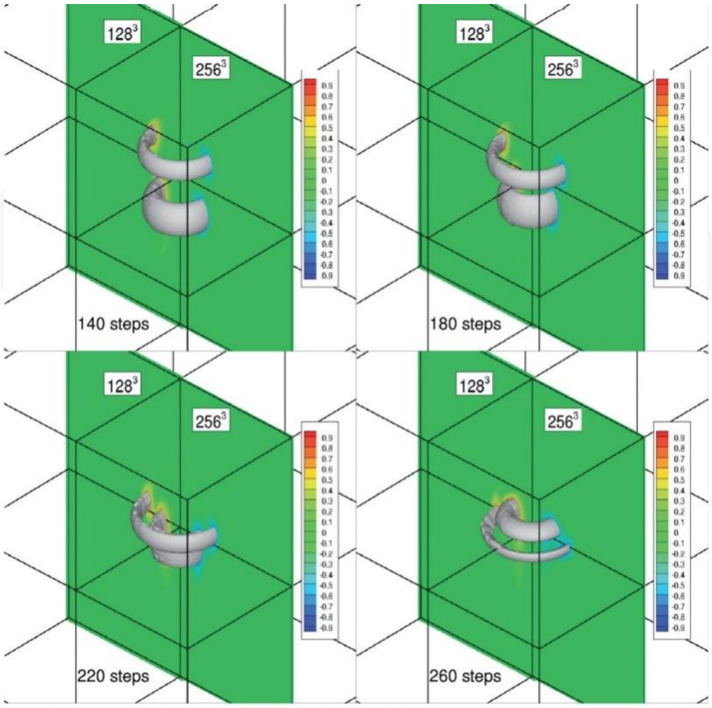
Vortex-in-cell simulation of leapfrogging vortex rings. Effect of mesh refinement is shown. "Noiseless" simulation using full QFT.

Fig.1



Vortex-in-cell simulation of leapfrogging vortex rings. For two different mesh resolutions, the effect of applying AQFT is shown ("k" limit is 5).

Fig. 2



Vortex-in-cell simulation of leapfrogging vortex rings. For 1283 mesh, AQFT ("k" limit 5) is used. For 2563 mesh, "k" limit in AQFT is 6.

Fig.3

Fig. 1, shows an example of a VIC simulation of two leapfrogging vortex rings, i.e., flow structures of fundamental importance in fluid mechanics. The lower vortex ring is stronger than the ring above it, and it will therefore convect upward faster, leading to the interaction of the vortex rings as shown. Results were compared for two different meshes, 128^3 and 256^3, to highlight the dependency of the solution on the chosen mesh size.

Results were obtained for noiseless QFT and as well as AQFT and were compared with "k" value to be 5 and showed that for the finer mesh this leads to unacceptable errors, while the coarser-mesh simulation still produces worthwhile results. When k limit was increased from 5 to 6 in Fig.3 , i.e., more controlled rotation gates are included in the AQFT circuit, the simulation on the finer mesh can also be made to produce similarly useful results showing the level till the approximation can be tolerated.

Quantum algorithm for discrete-velocity method

Discreet Velocity Method

In computational fluid dynamics, the most widely used methods involve solving the Navier-Stokes equations for a continuum fluid, i.e., where fluid density, velocity components, and energy in each location in the computational domain are to be found from conservation equations. The vortex-in-cell method shown above used vorticity rather than velocity. An alternative approach to the Navier-Stokes-based modeling is a description of the flow at a more detailed level, i.e., at the kinetic level. Instead of governing equations for mass, momentum, and energy conservation, the flow is now described by the Boltzmann equation governing a particle distribution function in state space (or 3D velocity space for a 3D monatomic gas flow) for each location in the considered domain.

Flow of Monoatomic gas is governed by Boltzmann equation-

$$\frac{\partial f}{\partial t} + \vec{c}\frac{\partial f}{\partial \vec{x}} = Q(f, f)$$

where, f is the distribution function, c is the velocity and Q is the collision term which represents the effect of collision between particles. But, due to higher computational cost the **collision-less Boltzmann equation** was used.

In the discrete-velocity method used here, the velocity space is discretized using a uniformly spaced Cartesian mesh.

The discrete-velocity approach used here has a number of characteristics facilitating a quantum implementation:

1. A uniformly spaced Cartesian mesh is used for the spatial discretization as well as for the discretization of the velocity space.

2. In case solid objects are present in the computational domain, these are rectangular, and its edges align with the mesh lines in the mesh. Specifically, solid bodies can be defined by “tagging” selected groups of cells in the mesh.

3. A constant velocity-space discretization is used in each point in space, i.e., the velocity-space boundaries defined earlier as well as the number of discrete velocities are identical in each cell.

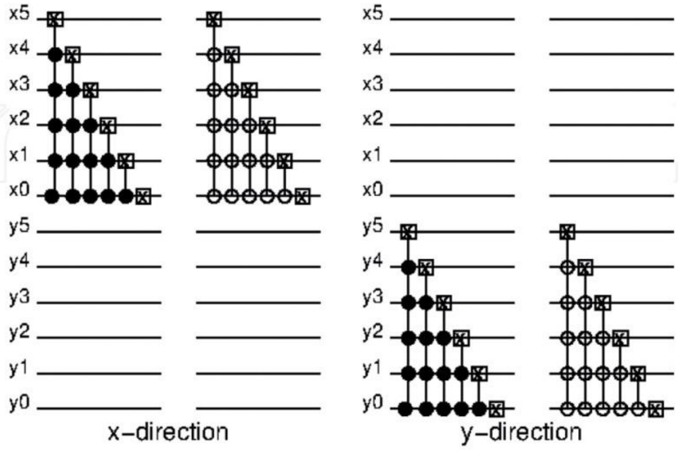
4. The convection part of the Boltzmann equation (i.e., the second term on the left-hand side in the shown equation) along with gas-solid interactions determines the time evolution of the distribution function in the absence of interparticle collisions.

5. The time-integration method used here is based on the reservoir technique, such that during the time integration the convection step always exactly involves the distribution function defined in a cell of computational mesh to move to a cell that is a nearest neighbor. This is commonly referred to as “streaming” of data.

For implementation, the collisionless Boltzmann equation, originally defined for 3D can be reduced to two kinetic equations for two reduced distribution function.

$$\frac{\partial f}{\partial t} + \vec{c}\frac{\partial f}{\partial \vec{x}} = 0; \frac{\partial g}{\partial t} + \vec{c}\frac{\partial g}{\partial \vec{x}} = 0$$

An efficient **Quantum circuit implementation** of the normal streaming operation in x- and y- direction on a Cartesian 2D mesh created as in the figure shown.

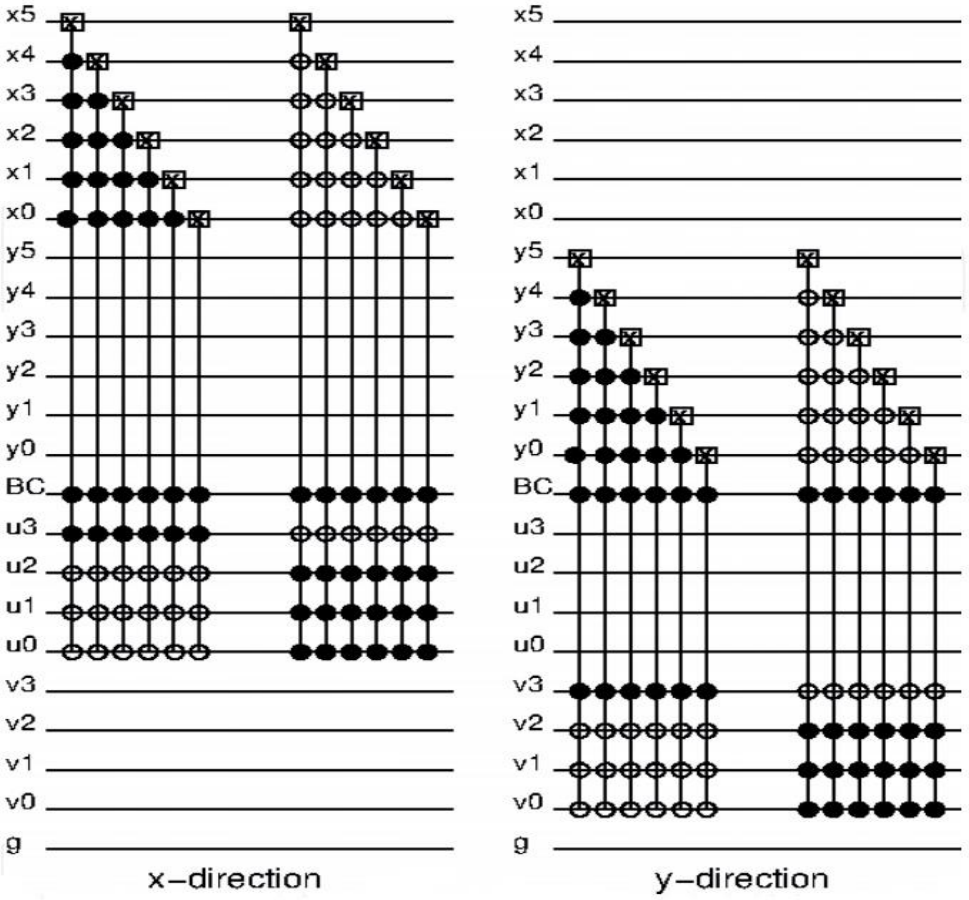


Quantum circuit implementation of “left streaming” and “right streaming” in x- and y-direction on a 64 × 64 Cartesian mesh.

Fig.4

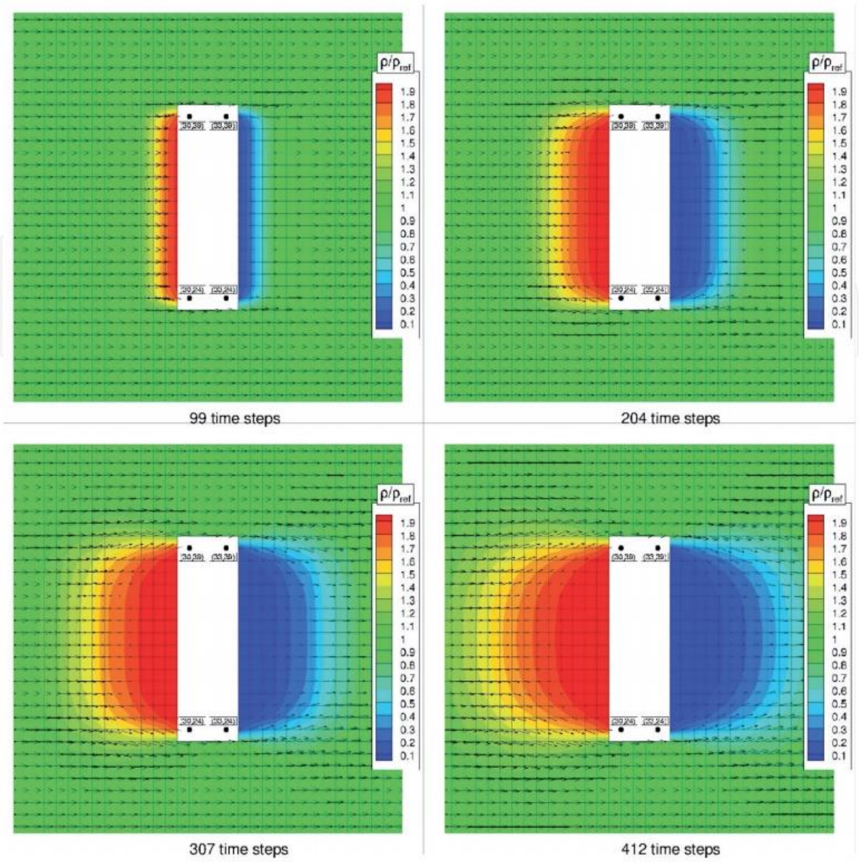
It can be seen that the figure shows how a 12 qubit register in Fig.4 is used for a discretized function with 6 qubits defining the indices of the 64 grid points in the x and y directions. The circuits with the filled circles define streaming to "right" neighbors, i.e., when each control qubits has the |1> state, the target qubit gets negated ("X" symbol used here). Similarly, the left streaming operation employs multiple-control NOT gates with target qubits being negated when each control qubit is in state |0>.

But, for **discreet velocity method** the circuit was extended to account for the storage of two discretized reduced distributions defined in discretized velocity space. The additional distribution function is accounted for using an additional qubit termed "g" in the following quantum circuit diagrams. The number of additional qubits needed for the discrete-velocity mesh clearly depends on the number of discrete velocities used. Finally, to account for solid objects, an additional qubit ("BC" in the diagrams) was used to set |1> to denote a cell within fluid and |0> for a cell within a solid. . For a 64 × 64 Cartesian mesh and a 16 × 16 discrete-velocity mesh, Fig. 5 shows the quantum circuit used to simulate the free-molecular flow around a rectangular body, for which the evolution of the flow field starting from an initial uniform flow is shown in Fig.6.



Quantum circuit implementation of streaming operations for discrete-velocity method (with 16 × 16 velocity mesh). Two-dimensional domain with 64 × 64 Cartesian mesh.

Fig. 5



99 time steps

204 time steps

307 time steps

412 time steps

Discrete-velocity simulation of Mach 2 flow around rectangular body. 64 × 64 Cartesian mesh, velocity-space mesh with 64 × 64 discrete velocities.

Fig. 6

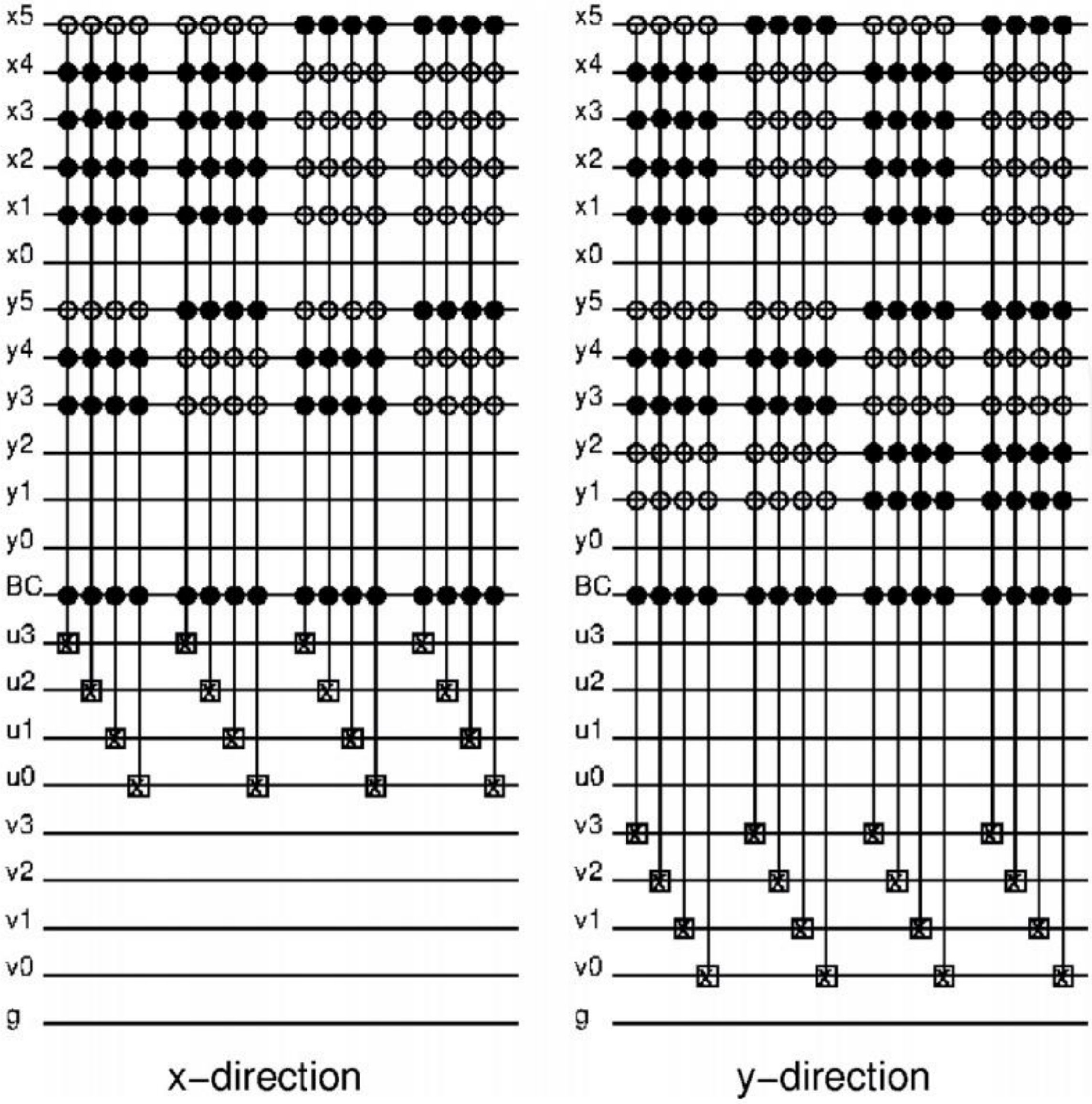
The key feature in the quantum circuits is the extended number of control qubits, i.e., beyond the checks on the qubits corresponding to spatial coordinates, control qubits also involves the "BC" qubit (fluid/solid flag) as well as the qubits related to the discrete-velocity indices. This least feature originates from the need to stream only data associated with selected discrete velocities in the used time-integration method.

Specular-Reflection Boundary Conditions

In the work shown, specular-reflection boundary conditions were assumed. This condition says that upon hitting a solid surface, a gas particle will bounce off this surface with the wall-normal velocity component effectively getting reversed and the tangential-flow component being preserved.

This boundary condition gradually makes the velocity vectors align with the solid wall such that there is no longer a flow into the solid body. This is the physically correct behavior.

In the quantum register, the indexing for the velocity-mesh data is designed such that a change of the sign of the discrete velocity implies a bit negation of qubits representing the index of the considered discrete velocity.



Quantum circuit implementation of specular-reflection boundary conditions for rectangular body. 64 × 64 Cartesian mesh, 16 × 16 discrete-velocity mesh.

Fig.7

In the case, only 16 × 16 discrete velocities were used for clarity. It can be seen that control qubits were used to "select" cells in space for which to apply the boundary condition. A further control qubit involves the “BC” qubit representing the solid/fluid flag. The negation operation (“X”) was applied to the qubits representing the velocity-mesh index to create the “change of sign” of the considered discrete-velocity data.

How AI & QC are changing/will change the face of CFD?

Increased number of Viable Diagrams

Growing the capability within an organization to use CFD in conjunction with QC could have breakthrough benefits in the long term as research department could make use of thousand simulations to develop better manufactured parts or to solve production problems that require deep study using CFD today. These could be optimized over several thousand scenarios, using more parameters, if the computation time was reduced using QC.

Because of computational power needed to run simulations, assumptions are made about some model inputs. This would be less relevant when using QC because more models could be run in a much shorter time. So, these inputs could be varied until the model is more accurate when compared to the real world. Then that model could be used to hone in on a better solution before starting trials in the plant.

Reduce Modelling Time

Solving computational fluid dynamics problems using traditional methods requires a lot of time and a lot of effort. For researchers who rely on data, and their interrelationships, running simulations to understand complex dependencies is an everyday task.

Combining AI and Quantum Computing will offer faster computing (e.g. architectures that are optimized for high volume, rapid, physics-based calculations) and have multiple effects on CFD modelling and use for manufacturing. AI models have developed to simulate the results of physics-based models with fair results, but a lot of training data is required to get AI models of comparable accuracy for a given problem. QC would not allow for multiple variations of the research goal to be simulated and compared in seconds instead of days or weeks.

In one example, a CFD simulation exercise was performed by Germany’s Renumic GmbH and UberCloud with the objective to decrease solving a fluid flow problem by 1000 times by using Artificial Neural Network (ANN) with the same accuracy as a traditional CFD solution process. In one of the exercises found, not surprisingly with large amounts of data, they had better accuracy, six times faster than the traditional process, with tens of thousands of samples available in hours rather than days. Add the capabilities of Quantum Computing and the time span could eventually be minutes. Thus Quantum Machine learning may help in developing much better result with much less time when served through sufficient amount of data.

Future Aspects

1) The need for further efficient quantum algorithms as well as a further understanding of how to apply the **Quantum coprocessor model** for this type of flow simulations is to be investigated.
2) The measurement-based extraction of classical information fundamentally changes the way quantum algorithms for CFD application will most likely be used.
3) Obtaining detailed information on the full flow field will be a challenge, so applications for which only certain characteristics of the solution are desired would present a good choice for future applications.
4) If Quantum Machine Learning can revolutionize CFD Problems.
5) Sensitivity Study of QFT based CFD Algorithms to know how much approximation can be useful for perfect results.
6) Implementing different solvers of CFD through Quantum Algorithms.
7) Fast and efficient error correction in results through QFT.

CONCLUSION-

Quantum Computing on computational fluid dynamics is still at its genesis and is going through to certain developments to implement the different quantum algorithms into solving the flow equations corresponding to different models which previously exist. In the near future, it is expected the application of quantum computing will follow the quantum coprocessor model, where parts of computation for which fast quantum algorithms exist are executed on quantum hardware.

In this report two different Quantum Algorithms were described. Beyond their very different areas of application, the key differences are the computational model with regard the quantum coprocessor model of quantum computing. The hybrid quantum/classical algorithm for the vortex-in-cell method involves repeated exchanges of information between classical and quantum hardware, i.e., at each time step in the time integration. In contrast, the quantum algorithm implementing a discrete-velocity method for kinetic flow modeling can be performed on the quantum processor for the duration of the simulation, with classical information exchange only required at the start and end of the simulation.

For the next report, the influence of Quantum Computing in Quantum Systems of CFD-for eg- Quantum Lattice gas Simulation will be included with new theoretical models which can be implemented for different solvers to get much less errors will be demonstrated that will be applicable in NISQ era of Quantum Computing.

REFERENCES:

1) Introduction to CFD Basics-Rajesh Bhaskaran, Lance Collins
2) The Present Status, Challenges, and Future Developments in Computational Fluid Dynamics- Antony Jameson
3) Quantum Algorithm for Fluid Simultions- Rene Steijl
4) Finding flows of a Navier–Stokes fluid through quantum computing.
5) Quantum Computing in the NISQ era and beyond, John Preskill

6) Ray, N., Banerjee, T., Nadiga, B., & Karra, S. (2019). Towards Solving the Navier-Stokes Equation on Quantum Computers. *UC Irvine*.

7) Steijl, R. and Barakos, G. N. (2018) Parallel evaluation of quantum algorithms for computational fluid dynamics. Computers and Fluids, 173, pp. 22-28. (doi:10.1016/j.compfluid.2018.03.080).

8) Quantum Computation of Fluid Dynamics, SACHIN S. BHARADWAJ and KATEPALLI R. SREENIVASAN

9) Fillion-Gourdeau, F. et al. "Algorithm for the solution of the Dirac equation on digital quantum computers." *Physical Review A* 95 (2017): 042343.