Quantum CFD: Utility Benchmarking

- Draft Proposal -

A. Gallo (PM & DevOps, GE Aerospace)
M. Alhawwary (CFD Methods, GE Aerospace), t.b.d. (CFD Methods), t.b.d. (DevOps), & others e.g. ORNL

Abstract—Industrial quantum utility for CFD in the coming years will depend on the arrival of fault-tolerant quantum computers at scale. In the meantime benchmarking open source quantum CFD algorithm implementations, and those already commercialized, on multiple simulators and quantum machines will provide valuable insights for future adoption, including a better understanding of which solutions best apply to which CFD cases. We propose a flexible 80/20 rule of thumb for assessing the maturity and readiness of quantum CFD implementations for industrial use cases, comparing the current performance of quantum algorithms in terms of fidelity and case scaling as compared to also evolving expectations of classical performance. We enumerate algorithms which have open source and vendor implementations and propose a cross-functional project for assessing their performance and scalability in today's immature quantum NISQ era. Using rigorous capture of case metadata, we incorporate the independent benchmarking activities of other external teams with our own complementary benchmarking efforts. We highlight the issues associated with benchmark repeatability and cross-platform consistency across diverse quantum hardware and software ecosystems, emphasizing the need for standardized evaluation protocols to ensure meaningful comparisons across platforms and over time.

1 Quantum Utility Benchmarking for CFD

1.1 Benchmarking Challenges

To consider a benchmark for quantum CFD, we need to consider what we are evaluating, and to what we are comparing. It would be typical to assess the speedup of an algorithm relative to some prior algorithm on some repeatable case. It would also be typical to compare the accuracy of the new solver relative to the prior accepted results.

In the case of quantum CFD, we intend to conceed the speed contest to classical solvers now and for the foreseeable future. There is no hardware roadmap which brings sufficient quantum volume to bear in the near-term to be used to solve the size of industrial problems currently solved by classical CFD entirely on a quantum computer, and therefore there is no basis for conducting a speedup contest at this time. Since sufficient "quantum volume" (itself a disputed term in the industry) cannot be brought to bear near-term, hybrid solutions like Variational Quantum Linear Solvers (VQLS) are used [1] where iterations between the quantum and classical nodes are orchestrated. This is most certainly not faster than a native classical solution due to at minimum the transport and other orchestration costs, which are all currently a subject of research interest especially as conducted by data centers hosting their own quantum computers in an HPC setting [2].

For accuracy, noisy near-term quantum computers will not be useful - only when sufficient qubits are available to allocate a good proportion to error mitigation will the remainder of "good" qubits be useful for improving the accuracy of the quantum algorithm. Meanwhile, our classical methods continue to improve, in size and granularity of the problem, accuracy, and speed to solution. We're less interested in how the quantum algorithm is improving over time relative to its own performance on some fixed reference, and more interested in a comparison of the quantum methods to our current state-of-the-art classical methods. In addition, the hardware will continue to evolve, and the software stacks which support it may as is common not maintain backward compatibility - it will become challenging to compare benchmark results even year-over-year.

The algorithms themselves present challenges to benchmarking given their active research status. The Harrow-Hassidim-Lloyd (HHL) algorithm [3] is normally run a single time in the literature, not iteratively over time steps as is typical in CFD solvers. Iterative methods are more robust to error, and can be designed to converge despite error in the solution at each step, as long as the error is bounded and does not bias the solution. Quantum linear solvers are being adapted to be used in this manner more consistent with CFD practice as shown in a 1D nozzle experiment [4] and with VQLS, but this is still an area of active research [5].

Incremental improvements in quantum algorithms may also change the benchmarking landscape over time. For example, if a quantum matrix conditioning algorithm is developed [6], or one which can be used as a pre-processor to a classical solver, this hybrid approach may be more effective than a pure quantum linear solver approach. A benchmarking framework needs to be flexible enough to accommodate such developments. Improvements in quantum algorithms can lead to improvements in classical, and vice versa, so the benchmarking framework needs to be dynamic and adaptable. The academic community has stood up algorithmic advantage trackers to monitor such fast-moving developments [7], [8].

In today's terms, quantum CFD fails the comparison with existing classical on all three measures: size, speed, and accuracy. If we conceed speed we can either conceed to classical entirely, or we can benchmark on accuracy and size, which today are so far out of the realm as to be not worthwhile to report. We will instead be interested in a *benchmark we can use to represent line-of-sight to near-term industrial utility*. We emphasize that the standard is *utility*, not *advantage*, and we will not engage at this time in the debate of whether quantum CFD can ever achieve practical advantage.

1.2 Benchmarking 80/20 Rule

We propose a **dynamic benchmarking rules** that monitors evolving quantum CFD methods against our continuously advancing classical state-of-the-art, focusing on accuracy and case size rather than speed. As a rule of thumb, *when quantum CFD algorithms can address* ***20% of a representative case size*** *with* ***80% of the desired accuracy****, industrial utility is in view for that case.*

Here the target case size is to be selected just-in-time and realistically, e.g. a small airfoil, or a small airfoil section, knowing that our largest HPC problems are still often only subsets of total system geometry. In other words, if the current classical case size is a full airfoil, quantum may be expected to address 20% of that geometry, with 80% of the current classical accuracy.

The benchmark will be re-evaluated periodically, perhaps annually or bi-annually, as both the classical *and* the quantum algorithm continue to improve. The frequency of evaluation should match the pace of advancement and anticipated proximity to the threshold.

As time goes on, it is expected that the number of algorithms, or variations on algorithms, to be evaluated will increase. Some algorithms may be pure quantum, others hybrid quantum-classical, and there may emerge other classifications. Some will be open source, some already are being commercialized. The use of a hybrid quantum algorithm, or quantum subroutine, may also be used in a future evaluation, for example in plugging in a quantum matrix conditioner prior to running a classical solver. Many variations on the use of quantum algorithms in whole or in part are possible. It will be the task of the research team to select the candidate or candidates for evaluation for any given iteration of the benchmark.

The benchmarking framework should be flexible enough to accommodate these variations. Over time, a suite of benchmark cases may be established, representing different CFD scenarios of interest to our business. These may be grouped into categories by algorithm type, application, and their relative performance tracked over time relative to each other, the classical state-of-the-art, and the 80/20 threshold. These types of ensemble comparisons are used in other domains, such as computational chemistry and the benchmarking of DFT algorithms and their various implementations [9]. Given the rapidly evolving landscape of quantum computing capabilities and classical CFD advancements, a single static benchmark is insufficient. Instead, we propose an ensemble of benchmarks that can adapt to new developments and provide a comprehensive view of progress in quantum CFD. Over time the list of candidate approaches may grow, or be winnowed down to a smaller set of leading contenders.

We propose an **ensemble of benchmarks** of various quantum CFD algorithms, their various leading implementations both open source and commercialized, over case scenarios of interest, continuously updated to reflect the evolving landscape of quantum computing capabilities and classical CFD advancements.

1.3 The Role of Software in Quantum Computing

By its very nature, quantum computing is multidisciplinary. Physics and engineering experts construct the hardware and networking of quantum computing, but discrete mathematics and software enter into error correction techniques beginning right on the device. There are pure domain experts such as in chemistry and material science, however, at the core of quantum computing applications which yield scientific or business results are software engineers, often with but sometimes without other specific domain expertise.

Its been suggested that fault tolerant quantum computers will usher in a new era of algorithm development not seen since the 1950 and 60s [10]. This development might not result in quantum methods gaining supremacy - it might result in improvements to classical methods which are informed by the quantum perspective, or developed as a result of chasing the quantum promise. Already tracking sites have been established for baselining algorithms regardless of their quantum or classical origin [11].

Software for quantum computing comes in several types:

Quantum circuit authoring frameworks - libraries to construct quantum circuits, often with simulators to test them on classical hardware. Examples include IBM Qiskit, Google Cirq, Xanadu PennyLane, Qrisp, Munich Quantum Software Stack, and others. Debugging tools are emerging.

Quantum hardware control software - low level software to control the quantum hardware, often including pulse-level control for fine-tuning qubit operations.

Quantum algorithm libraries - collections of pre-built quantum algorithms for specific applications, such as optimization, chemistry simulations, or machine learning.

Quantum cloud platforms - services that provide access to quantum hardware and simulators via the cloud, often with integrated development environments, workflow orchestration, and co-scheduling with classical HPC resources. Better metrics tools which permit profiling classical / GPU and QPU resources are emerging.

Domain-specific quantum software - applications tailored to specific industries or scientific domains, such as quantum CFD solvers or quantum chemistry packages.

It is this last category which is of most interest to us, however, the other categories are important enablers and may be required elements of a complete quantum CFD solution. Their use, or not, in conjunction with the offerings of CFD-specific vendors may be complementary or competitive.

1.4 Quantum CFD Algorithms & Implementations

A range of current open source and vendor offerings will be considered for evaluation and benchmarking. Implementations may not exist for all of these algorithms, and the scope of this project *does **not** include* providing implementations where they do not exist.

An aspect of the proposed project is to categorize quantum CFD algorithms when those assignments are useful for the overall tracking of the benchmarking framework, and to evolve these categories over time to align with the specific CFD use cases which best apply.

1.5 Partial List of Algorithms & Variations

A non-exhaustive list of algorithms and variations includes:

Harrow-Hassidim-Lloyd - quantum linear system algorithms for solving discretized PDEs, including HHL and related variants, such as LuGO for reducing the cost of the Quantum Phase Estimation (QPE) portion of HHL [12].

KU-SVD - a practical quantum solver for multidimensional PDEs developed at the University of Kansas, using singular value decomposition techniques to address the linear systems arising from PDE discretization [13].

Quantum LBM - quantum lattice Boltzmann methods for fluid dynamics simulation, focusing on quantum speedup for collision and streaming operations, with potential applications to turbulence modeling and diffusion problems.

Variational Quantum Algorithms - hybrid quantum-classical approaches including Variational Quantum Eigensolvers (VQE), Variational Quantum Linear Solvers (VQLS), Quantum Approximate Optimization Algorithm (QAOA) implementations, and others for solving optimization and eigenvalue problems present in CFD.

Quantum Machine Learning - quantum-enhanced machine learning techniques [14] for data-driven CFD modeling [15], including quantum neural networks, physics-informed neural networks, and quantum support vector machines [16].

Schrodingerization - quantum algorithms for solving the time-dependent and time-independent Schrodinger equations with potential applications to quantum fluid dynamics simulations, particularly in quantum hydrodynamics and quantum transport phenomena.

Other Algorithms - miscellaneous quantum algorithms for CFD applications, including for example application of the quantum max cut algorithm in mesh pre-processing.

1.6 Current List of Quantum CFD Vendors

The following are a sampling of some of the software vendors attempting to commercialize quantum CFD solutions. This is not necessarily an exhaustive list, but rather a snapshot of the current commercial landscape as of late 2025, and one which might inform our benchmarking plans.

BosonQ Psi (BQP) takes a pragmatic approach to quantum-enhanced simulation, developing quantum-inspired and quantum algorithms that can integrate into existing simulation workflows. The company claims significant performance advantages compared to classical approaches, their technical innovations include Quantum-Assisted Physics-Informed Neural Networks achieving speed-up for CFD training, and a collaboration with Classiq and NVIDIA that achieved a circuit compression improvement. *Contact: Abhishek Chopra, CEO, abhishekchopra@bqpsim.com*

QubitSolve represents a focused approach to quantum CFD, developing a dedicated quantum computing solution specifically designed to tackle the Navier-Stokes equations that govern fluid behavior. Their approach is variational, hybrid quantum-classical. They acknowledge that the current crossover point where quantum performance surpasses classical methods remains too large for practical industrial applications. QubitSolve is actively working to shrink this crossover point with an ambitious target of reaching practical utility by 2027. *Contact: M. Syamlal, CEO, msyamlal@qubitsolve.com*

Ansys maintains its position as a traditional leader in classical CFD simulation software and is actively engaged in researching quantum algorithms for partial differential equations and quantum Lattice Boltzmann Methods. The company has adopted NVIDIA CUDA-Q. *Contact: t.b.d.*

ColibriTD has developed QUICK-PDE, a Qiskit package for solving partial differential equations, starting with the inviscid Burgers' equation and materials deformation applications for hypo-elastic one-dimensional tensile testing. The company's approach is based on their H-DES algorithm, which encodes trial solutions as linear combinations of orthogonal functions, typically Chebyshev polynomials, parametrized by the angles of a Variable Quantum Circuit. The algorithm operates through a hybrid loop where the ansatz generates a state encoding the function, which is evaluated by observables that allow function evaluation at all points, followed by loss function evaluation in which differential equations are encoded and angle fine-tuning until satisfactory results are achieved. ColibriTD's implementation leverages noise mitigation within the optimization process by stacking multiple identical circuits and evaluating identical observables on different qubits throughout one large circuit, relying on a noise learner method to significantly reduce the required number of shots. *Contact: Henri de Boutray, henri.de.boutray@colibritd.com*

Quanscient operates in the broader multi-physics simulation space with a cloud-based platform and workflow management for AI model training from simulation. Their CFD work is centered around the quantum LBM, researching methods to scale their quantum CFD simulations to larger grids and more complex geometries with a focus on optimizing circuit depth and error mitigation techniques.

Contact: t.b.d.

1.6.1 Question of NDAs

NDAs can be established to probe specific vendor capabilities to qualify them for future benchmarking exercises. However, NDA are problematic in a multi-party project as they must bind all parties under similar terms, and the findings are of course not openly publishable. Thus we recommend using NDAs only in a limited capacity to probe vendor capabilities in the out years.

It's also worth noting that several of these vendors are partners of IBM Quantum and offer their products as part of the Qiskit Functions catalog [17]. This is one means of obtaining access.

2 Project Proposal

2.1 Experiment: Case x Code x Backend

The project proposal focuses on establishing a comprehensive benchmarking framework for quantum CFD algorithms. This framework will systematically evaluate and compare quantum algorithms for computational fluid dynamics across multiple dimensions: code implementation, CFD case, and quantum backend towards exceeding the proposed 80/20 utility rule. The evaluation will be encoded with repeatability in mind, while recognizing the problems inherent with that approach.

CFD cases are likely to be very small at this stage of the noisy intermediate-scale quantum (NISQ) era, and we expect the set of cases to evolve in complexity and size over successive runs of the evaluation, over timeframes measured in months and years, towards the anticipated arrival of utility using the 80/20 rule as the guide. In the current iteration of the evaluation we expect to not meet the 20% case size criterion, but we may find ourselves in range of the 80% accuracy threshold. Each case description carries with it quantified standards for the 80/20 rule, for that case as used in useful industrial applications. In case specification, dimension does not matter much, what matters is complexity of physics, increasing as cases become more amenable to maturing quantum hardware.

Quantum backend selection is a practical concern. Simulators are available, including those modeled after specific physical backends, but these are limited by the classical compute power of the machine hosting the quantum simulation. This may dramatically limit the number of qubits and circuit depth, and likely result in the case failing the 80/20 rule on the basis of case size. Real quantum computers are also available, though they come with their own set of constraints and limitations - number of qubits, qubit coherency time relative to circuit depth, and cost considerations. As the benchmarking effort towards utility is a multi-year endeavor, we expect that current quantum hardware will be obsolete and unavailable for use in the later stages of the project. This limits the longevity of specific experiments, but not necessarily that class of experiments (e.g. 1D nozzle cases).

2.2 Case Set

Experiments are defined by the combination of a specific CFD case specification, a coded implementation, and some quantum backend, perhaps a simulator. Implementations for the algorithms, and those vendor-provided, will be identified as part of the project plan, below 2.7. Backends are similar - use of both simulators and real backends each pose their own issues to access and usage, and the specific enumeration will be made as part of the plan itself.

Here we enumerate some of the potential CFD cases which may be applied to experiment construction. This is also not an exhaustive list, to be similarly formalized as part of the project itself. Examples include:

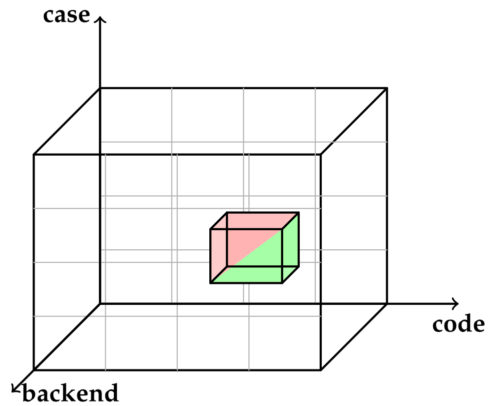

Fig. 1. Conceptual benchmark cube: tracking by exemplar CFD case (e.g. "Hele-Shaw 1D supersonic divergent nozzle 5x5 grid"), coded implementation (e.g. "HHL with LuGO"), as run on one or more target quantum backends (e.g. "IBM Kingston"). For a given (case, code, backend) experiment we notate the performance relative to the two parts of the 80/20 rule - accuracy, and case size relative to industrial utility for that experiment. Each part is graded pass/fail separately.

- Hele-Shaw flow in a thin rectangular box
- 1D supersonic divergent nozzle
- 2D irrotational flow past a {cylinder, airfoil}.
- 2D {laminar, invicid} flow in a {pipe, airfoil}
- Advection-diffusion scenarios
- Turbulent boundary layer transitions
- Shock wave interactions
- Steady and unsteady cases
- Compressible and incompressible flow

CFD cases can be modeled simply with the Poisson equations, or in increasingly complicated forms using the Burgers and Euler's equations, and the Navier-Stokes equations. Case selection can also be informed by other CFD industry initiatives [18], [19].

2.3 Partial Case x Code Mapping

Considering just the CFD cases and scenarios and the known implementations, and setting aside the quantum backend resource provisioning, we can begin to see the outlines of the experiment execution plan.

TABLE 1
Code x Case mapping (partial and abbreviated). We expect to develop a more complex matrix as part of the effort. Over time, this matrix may evolve as the market evolves. Here we show instances where a single CFD case can be approached using more than one code solver as well as conceptual instances where no solver has yet been implemented.

Code	HeleShaw	1D noz	2D pipe	AvecDif
ORNL HHL	x			
ORNL LuGO	x			
VQLS	x			
GE HHL		x		
KU SVD	x			
RPI QLBM				x
BQP				x
ColibriTD				x

2.4 Test Harness

Quantum applications nearly never run solely on the quantum computer - circuit construction is at minimum classical, so is post-processing, and some algorithms are iteratively hybrid quantum-classical. Therefore each experiment is implemented by a workflow, which includes classical preprocessing, quantum execution, and potentially also classical post-processing steps. For consistency, a standard workflow tool, but not a standard workflow, will attempt to be used for all experiments. This will hopefully aid repeatability and maintainability.

Pre-processing like circuit transpilation and optimization, and circuit execution on quantum backends come with their own sets of parameters. For example, the size of the problem forces a minimum shot count - there must be sufficient shots to show signal over noise. Additionally, quantum error correction and noise characteristics can significantly impact the reliability of results, making it essential to account for these factors in the benchmarking framework to ensure meaningful comparisons across different quantum platforms. Metadata about the specifics of the execution parameters and environment must be captured as part of execution, and the selected workflow tooling should permit this easily. Containers of some form might also be useful here. A specific list of test harness tool requirements will be formed as part of the project.

It may be useful to publish the requirements of the test harness as some of the experiments defined in the plan may be executed by parties not directly related to, but tangential to, this effort. For example, where a code is proprietary and access cannot be secured, the vendor may execute the test experiment, and we would require certain reporting to fit the expected results framework.

2.5 Team

As seems the case in quantum computing, the proposed team is inherently multi- and cross-functional, consisting of classical and/or quantum computing experts and domain specialists who understand both quantum algorithms and CFD applications. Persons can play more than one role, and more than one person can play a given role. Some team roles include:

- CFD domain expert - maintains the scientific integrity of the project, defines the test cases and validation criteria, analyzes experiment results
- DevOps - selects tooling, authors experiment workflows and schedules them for execution, manages access to infrastructure
- Vendor and other external liaison - coordinator for external hardware and software partners and vendors
- Project manager - communication, team coordination, timeline and tracking

For resiliency the ideal team would consist of more than one CFD expert, and more than one DevOps engineer. The team would span the interests of industry, government, and academia. From a CFD domain perspective, there may be a tendency to prefer a focus on and experts from an aerospace domain, but in light of the estimated roadmap to quantum utility for aerospace, other domains should also be considered.

2.6 Potential Outcomes

Some potential outcomes which result from the data gathered in these experiment trials might include:

- Identification of quantum utility regions for specific CFD cases
- Applicability of specific algorithms or vendor solutions to specific CFD case types
- Performance scaling models for quantum algorithms vs classical baselines
- Error mitigation strategy effectiveness assessment
- Error impact on specific algorithms or implementations
- Impact of qubit $\{$technology, connectivity$\}$ on CFD utility
- Resource requirement projections for larger-scale CFD simulations
- Efficacy of the selected workflow tooling for reproducible benchmarking
- Recommendations for standardized benchmark reporting across vendors
- Recommendations for future CFD benchmarking efforts and supporting tool improvements

2.7 Plan

Given industrial utility requirements, we will develop a dynamic plan that evolves per year as tools and hardware improve. While the intention is to perform most of the design work in the first year and hand-off, the overall plan will be updated annually to reflect advances in quantum computing technology and algorithm development. As utility comes into view based on the 80/20 rule, we may elect to run the benchmarks more frequently than annual. Here coding for reuse, as best as able, is key to lowering these costs over time, enabling a broader set of users and maintainers.

Based on individual and public vendor roadmaps which target fault tolerant quantum computers at scale for the 2028-29 timeframe, a plausible benchmarking schedule over the coming years might be:

TABLE 2
Dynamic CFD benchmarking timeline (proposed)

Year	Task
2026	Establish benchmark framework
2027	Annual benchmarking
2028-29	Bi-annual benchmarking
2030	Potential CFD utility?

In the planning stage, the following tasks are an approximate plan of action:

1) **Down-select and define CFD cases in scope for the current evaluation.** For each, define all CFD case parameters, and the threshold of success in achieving industrial utility using the 80/20 rule. Understand that this list may evolve over time as we gain more insight into viable case sizes for quantum utility, and the current state-of-the-art hardware improves.

2) **Identify available compute resources.** This includes classical compute for pre- and post-processing and the running of simulators, and quantum backends with specific specifications and availability. Leverage free and available resources. Where needed, identify funding sources to secure additional resources.
3) **Select from available implementations.** Without constructing new ones, identify the open source and vendor offerings which are available to be evaluated. Not all implementations may be suitable for all CFD cases or all target backends.
4) **Enumerate the test plan.** Based on the identified cases, codes, and backends, enumerate the experiments in the test plan.
5) **Select or design the test harness.** With an eye towards the often evasive reuse and repeatability, identify the test tooling which will be used to conduct the experiments. The harness must be able to capture case metadata, including but not limited to the runtimes of workflow steps and the configuration (e.g. coupling map) and condition (e.g. gate fidelities) of the specific quantum backend at the time of execution.
6) **Execute the experiments.** Document the results with detailed metadata about the experiment, the runtime environment (e.g. code and library versions, current backend noise profile, execution elapsed times, etc.). Ideally the test harness captures this information.
7) **Report.** Publish results.
8) **Plan for the next iteration.** Quantum utility for CFD is projected to be some years away. These evaluations will be re-conducted periodically, with experiments altered, added, or removed, in accordance with the current state of the quantum hardware and software landscape.

As a draft timeline for the current year 2026, the following is proposed:

- 1Q: Initiation, team assembly, case selection, experiment design, tool selection, resource identification
- 2Q: Execute experiments
- 3Q: Execute experiments, analyze results
- 4Q: Report results, discuss future iterations

It's understood that the 2026 team may differ from that in future years. The intention would be to have a paper for publication prepared by the end of 4Q26.

References

[1] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. J. Coles, "Variational quantum linear solver," *Quantum*, vol. 7, p. 1188, Nov. 2023. [Online]. Available: http://dx.doi.org/10.22331/q-2023-11-22-1188

[2] U. Bacher, M. Birmingham, C. D. Carothers, A. Damin, C. D. G. Calaza, A. K. Karnad, S. Mensa, M. Moreau, A. Nober, M. Ohtani, M. Rossmannek, P. Rubin, M. E. Sahin, O. Wallis, A. Shehata, I. Sitdikov, and A. Wennersteen, "Quantum resources in resource management systems," 2025. [Online]. Available: https://arxiv.org/abs/2506.10052

[3] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Physical Review Letters*, vol. 103, no. 15, Oct. 2009. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.103.150502

[4] M. Alhawwary and A. Gallo, 2025, unpublished work, GE internal report.

[5] A. Bengoechea *et al.*, "Quantum time-marching algorithms for solving linear transport problems including boundary conditions," 2025. [Online]. Available: https://arxiv.org/abs/2511.04271

[6] J. Lapworth and C. Sünderhauf, "Preconditioned block encodings for quantum linear systems," https://arxiv.org/abs/2502.20908, 2025, preprint.

[7] Quantum Advantage Tracker. (2025) Quantum advantage tracker. [Online]. Available: https://quantum-advantage-tracker.github.io/

[8] T. Koch, D. E. B. Neira, Y. Chen, G. Cortiana, D. J. Egger, R. Heese, N. N. Hegade, A. G. Cadavid, R. Huang, T. Itoko, T. Kleinert, P. M. Xavier, N. Mohseni, J. A. Montanez-Barrera, K. Nakano, G. Nannicini, C. O'Meara, J. Pauckert, M. Proissl, A. Ramesh, M. Schicker, N. Shimada, M. Takeori, V. Valls, D. V. Bulck, S. Woerner, and C. Zoufal, "Quantum optimization benchmarking library - the intractable decathlon," 2025. [Online]. Available: https://arxiv.org/abs/2504.03832

[9] K. Lejaeghere, G. Bihlmayer, T. Björkman, P. Blaha, S. Blügel, V. Blum, D. Caliste, I. Castelli, S. Clark, A. Corso, S. de Gironcoli, T. Deutsch, J. Dewhurst, I. Marco, C. Draxl, M. Dulak, O. Eriksson, J. Flores-Livas, K. Garrity, and S. Cottenier, "Reproducibility in density functional theory calculations of solids," 03 2016.

[10] J. Gambetta, 2025, spoken remarks, IBM developer conference, November 2025, Atlanta, GA.

[11] Quantum Advantage Tracker. (2025) Quantum advantage tracker. [Online]. Available: https://quantum-advantage-tracker.github.io/

[12] C. Lu, M. Meena, and K. Gottiparthi, "Lugo: an enhanced quantum phase estimation implementation," https://arxiv.org/abs/2503.15439, 2025.

[13] M. Chaudhary, K. El-Araby, A. Nobel, I. Islam, M. Singh, and S. Ogundele, "A practical quantum solver for multidimensional partial differential equations," in *Proceedings of the SC '25 Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC Workshops '25. New York, NY, USA: Association for Computing Machinery, Nov. 2025, pp. 1916–1925.

[14] S. Y. Chang and M. Cerezo, "A primer on quantum machine learning," https://arxiv.org/abs/2511.15969, 2025, preprint.

[15] J. Amaral *et al.*, "Quantum machine learning and quantum-inspired methods applied to computational fluid dynamics: a short review," https://arxiv.org/pdf/2510.14099, 2025, preprint.

[16] S. Lloyd *et al.*, "Quantum support vector machine for big data classification," https://arxiv.org/abs/1307.0471, 2013, preprint.

[17] IBM Quantum. (2025) Ibm cloud platform, qiskit functions. [Online]. Available: https://quantum.ibm.com/services/functions

[18] C. Consortium, "Cfd2030 report," http://cfd2030.com/report/CFD_Vision_2030_Roadmap_Update_2024-FINAL.pdf, 2024.

[19] DARPA QBI Team, "Darpa quantum benchmarking initiative," https://www.darpa.mil/program/quantum-benchmarking-initiative, 2025.