

Article

A Cost-Efficient Approach towards Computational Fluid Dynamics Simulations on Quantum Devices

Szabolcs Jóczik ^{1,2} , Zoltán Zimborás ^{1,3} , Tamás Majoros ²  and Attila Kiss ^{1,4,*} 

¹ Faculty of Informatics, ELTE Eötvös Loránd University, 1117 Budapest, Hungary; joczikszabi@inf.elte.hu (S.J.); zimboras.zoltan@wigner.hu (Z.Z.)

² Robert Bosch, 1103 Budapest, Hungary; Tamas.Majoros@hu.bosch.com

³ Wigner Research Centre for Physics of the Hungarian Academy of Sciences, 1121 Budapest, Hungary

⁴ Department of Informatics, J. Selye University, 945 01 Komárno, Slovakia

* Correspondence: kiss@inf.elte.hu

Abstract: Numerical simulations of physical systems are found in many industries, as they currently play a crucial role in product development. There are many numerical methods for solving differential equations that describe the underlying physics behind the mathematical models in the simulation, among which, the finite element method (FEM) is one of the most commonly used. Although in many applications the FEM seems to provide an acceptable solution to the problem, there are still many complex real-life processes that can be challenging to simulate numerically due to their complexity and large size. Recently, there has been a shift in research towards efficiently applying quantum algorithms in finite element analysis (FEA), as the potential and speedup that they could offer have been shown, but little to no effort has been made towards the applicability and cost efficiency of these algorithms in real-world quantum devices. In this paper, we propose a cost-efficient method for applying quantum algorithms in FEA for industrial problems post-processed by classical algorithms in order to address the limitations of available quantum hardware and their cost when accessing them through different cloud-based services. We carry this out by approximating the solution of the initially large system with a suitable quantum algorithm and using the obtained solutions to generate a set of reduced-order models (ROMs) that are much smaller in complexity and size than the original model. This allows the simulation of the original model with different parameter sets and excitations to be run efficiently on classical computers without having the need to access quantum subroutines again. This way, we have reduced the usage of quantum hardware (and thus the development cost) while still taking advantage of its quantum speedup.



Citation: Jóczik, S.; Zimborás, Z.; Majoros, T.; Kiss, A. A Cost-Efficient Approach towards Computational Fluid Dynamics Simulations on Quantum Devices. *Appl. Sci.* **2022**, *12*, 2873. <https://doi.org/10.3390/app12062873>

Academic Editors: Cesare Biserni and N.C. Markatos

Received: 31 December 2021

Accepted: 8 March 2022

Published: 10 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The aim of numerical simulations is to approximate the behavior of physical systems on a computer with high precision. It is now a well established methodology that is widely used in many fields, such as solid mechanics, thermodynamics, computational fluid dynamics and chemistry, for a wide range of industries that require accurate solutions. It plays a crucial part in the designing phase of product development, where numerical simulations are carried out in advance for virtual prototyping, prior testing and optimization for time and cost efficiency. Numerical simulations can give us insights into understanding component and system behaviors for large-scale complex applications under extreme conditions and use cases that could be impossible to test physically beforehand. The range of application areas is growing constantly, with some examples being stress-strain simulations estimating the crash worthiness of automobile parts [1], stress analyses of high speed helical gears to reduce failures and for optimal design [2] and accurately predicting and confirming vehicle flight characteristics for Boeing commercial airplanes [3].

Among all the available numerical methods, the most commonly used ones are the finite difference method (FDM) [4,5], the finite element method (FEM) [6–8] and the finite volume method (FVM) [9–11]. These methods approximately solve differential equations that describe the underlying physics behind the mathematical models in the simulation. It works by discretizing the domain and converting the differential equations into a (typically large) set of linear equations. Although solving these large systems of linear equations with high performance computing (HPC) seems to be acceptable in many use cases, there are still many real-life processes that are difficult to simulate accurately due to their complexity and large size. Turbulent flows are examples of such complex systems that require more advanced methods to approach and to predict their behaviors [12–14].

Model order reduction (MOR) is a method that aims to reduce the computational complexity of large-scale dynamical and control systems. The main motivation behind MOR techniques is to remove the features and information of the full-order model (FOM) that are less important in terms of a given application while still maintaining a relatively high accuracy. By reducing the state-space dimension of the model (or degrees of freedom), an approximation to the original model is computed, which is commonly referred to as a reduced-order model (ROM) [15–18]. ROMs are small in complexity and cheap in terms of computational time; thus, they can be effectively applied in the early stages of product development, as in conceptual design, virtual prototyping and optimization (such as in naval shape design [19] and wind-driven ocean flows [20]), where highly accurate results are not yet desired. This allows for experimenting with different geometries and parameters under certain scenarios in a fast, iterative process before a high-order simulation is carried out for more accurate results and validation purposes. Reduced-order models are also extremely useful in settings where it is often unfeasible to perform numerical simulations using the complete full-order model due to limitations in computational resources or the requirements of the simulations setting. This is often the case in real-time simulation settings where a large number of simulations need to be performed [21,22]. For more applications of MOR in fields such as engineering, electromagnetics and neuroscience, refer to [23]. Most MOR methods can be classified as projection- or truncation-based techniques. Projection-based MOR techniques aim to lower the computational complexity by finding an appropriate space of reduced dimension where the high dimensional state-space can be projected onto and approximated by. Truncation-based reduction techniques tend to preserve the input–output dynamics of a system for a certain set of output states while removing any unnecessary details of the system that are irrelevant for the calculation of that given output set [24]. Lastly, there is an increasing number of work towards enhancing deep-learning for model-order reduction where traditional MOR algorithms could be problematic in capturing the nonlinear input–output dynamics of the system due to the strong variability characterizing the solution set [25–27].

Recently, researchers have been focusing on applying quantum computing in FEM and FVM [28,29], as the advantages that it could provide by exploiting its potential and computational speedup, compared to classical algorithms, have been shown. Nevertheless, little to no effort has been made towards their applicability and cost efficiency when used in real-world settings. In such cases, numerical simulations might need to be carried out numerous times with different parameters in product design, optimization and prior testing. As quantum hardware is mostly available through cloud services, each carried out simulation would increase the expenses and development costs, which could easily add up to the tens of thousands of dollars range in even small-complexity simulation problems. This could cancel out all of its benefits against classical computing for many small-to-medium-sized businesses, where a decision would have to be made between a faster development time and lower development cost.

In this paper, we propose a method for reducing the simulation time in computational fluid dynamics applications while also taking the development costs into account. We carry this out by combining a suitable quantum subroutine for solving the initially large system and post-processing the obtained solutions with classical MOR techniques to generate reduced-order models. The reduced models can then be used for effectively carrying out

simulations of the original system with different excitations and with a high accuracy using classical computers. This way, we can take advantage of the quantum speedup, but the need for using a quantum hardware is also minimized. We strongly believe that this can be a crucial factor in the near future, when quantum computing is starting to be used more and more often for industrial problems, but the available hardware is still very limited and expensive. Minimizing the number of jobs submitted on quantum hardware by cleverly combining quantum subroutines with classical post-processing techniques could save valuable resources and reduce the cost for any industries planning to invest in quantum computing in the upcoming decade.

2. Related Works

The combination of quantum computing and finite element analysis holds great potential, especially in such cases where current classical computers are struggling to solve complex systems accurately in a relatively fast time, such as in computational fluid dynamics (CFD) applications or tightly coupled multiphysics systems. Nevertheless, this field has been mostly ignored before, but has been gaining more and more attention lately [28,30,31]. Since the output from FEM is often large-sized matrices that are produced algorithmically, it would be a great first step to find quantum algorithms that can solve a system of linear equations in the form $Ax = b$ with exponential speedup. The FEM also leads to sparse system matrices, which is a useful and often required property to reach quantum speedup via the quantum solver algorithm. Lastly, as mentioned in the introduction, FEM has a wide range of industrial applications, such as solid mechanics, thermodynamics, computational fluid mechanics, chemistry and many more. Thus, any achievable speedup can mean a huge difference and may have a large impact on the industry.

A quantum algorithm for solving large systems of linear equations was first introduced by Harrow, Hassidim and Lloyd [32], and was later improved by Childs [33]. In these papers, the quantum speedup in solving a linear system of equations is claimed to be exponential. However, there are challenges in efficiently preparing the classical input parameters and measuring the output states at each time step, as these two operations can cancel out the exponential speedup that the algorithm offers, as was later proven by Montanaro and Pallister [28]. These problems need to be faced by any quantum algorithms that wish to have an exponential speedup compared to their classical counterparts. Nevertheless, there are new research activities mainly focused on solving non-linear systems of equations that claim to have found a workaround to the previously mentioned problems and have reached an exponential speedup in large scale systems [30,31,34,35].

In a recently published paper, Lloyd et al. [30] proposed a method where the number of required resources scale quadratically and logarithmically in the integration time and in the dimension of space-time, respectively, thus preserving the exponential speedup of the solver. In a separate study by Childs [33], a different approach was taken, where they used Carleman linearization to transform nonlinear problems into a system of linear equations. This system of linear equations consists of an infinite number of equations that have to be carefully truncated and reduced to a finite set to be applicable in real-world settings. In another recent paper by Chen et al. [29], an alternative approach was proposed by taking advantage of quantum RAM (QRAM) in order to preserve the quantum speedup offered by the HHL algorithm. Although all of these results show great potential in the field of numerical simulation for a wide range of applications, such as in CFD, these methods will likely be refined, worked upon and used to develop new methods in the upcoming years, before a suitable quantum hardware is available.

In this paper, we propose a method for applying the previously mentioned and other newly discovered quantum algorithms for real-life problems by taking advantage of their quantum speedup, but also taking the limited availability of quantum computers and their costs into account. We carry this out by minimizing the need to use quantum algorithms for solving PDEs by post-processing the solutions obtained from a few critical initial cases and creating ROMs, which can be efficiently solved on classical computers afterwards.

with different parameter sets. We solely focus on the FEM as our discretization method and will introduce it in Section 3. Nevertheless, the method we describe in this paper is generalizable in a sense that FEM can be replaced by other similar methods, such as FVM [36,37].

3. Preliminaries

3.1. The Basic Concepts of Finite Element Method

Numerical simulations of real-life processes cannot be carried out unless the corresponding mathematical model of the physical system is defined along with the underlying physical laws that govern the system. These laws for space- and time-dependent problems can be expressed in terms of partial differential equations (PDEs). In most cases, however, these PDEs are extremely difficult if not impossible to solve analytically. Therefore, an approximation often has to be made that can be produced using different types of discretization methods, such as the FEM (as seen in Figure 1), FVM and FD. In the case of CFD problems, FVM is the most commonly used scheme, as it is a locally conservative method, meaning that the conservation laws are satisfied on every cell in the mesh. In FEM, the conservation laws are only satisfied globally. Local conservation is an attractive feature, especially in the case of the Navier–Stokes equations; however, FEM is more flexible and generalizable to other problems and use cases. Thus, in this paper, we decided to use FEM as our main discretization method. For more details regarding the differences between FEM and FVM, refer to [38,39].

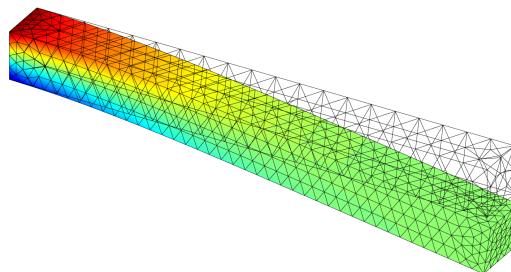


Figure 1. The finite element mesh discretizes the domain into a finite set of smaller subdomains, which are called elements. Over these elements, a set of basis functions are defined that approximately represent the PDEs that govern the system. As we refine the mesh by increasing the number of elements, the approximation will converge to the true solution.

The main features of FEM can be best explained using a simple example, but, for a more in-depth understanding, refer to [6–8]. For this reason, let us take a look at a heat conduction problem inside a thin metal rod of length L :

$$\text{differential equation} \quad -\lambda T''(x) = f(x) \quad \forall x \in (0, L), \quad (1)$$

$$\text{boundary conditions} \quad \begin{cases} T(x=0) = 0 \\ T'(x=L) = 0. \end{cases} \quad (2)$$

Here, λ denotes the thermal conductivity, $T(x)$ denotes the temperature at point x and f is the rate of heat generation per unit volume ($[f(x)] = W/m^3$). The left end of the rod is kept at constant temperature level 0, whereas, at the right end, no heat is allowed to enter or leave. The next step is to multiply both sides of Equation (1) with a so-called test function v and integrate both sides over the whole domain $\Omega = (0, L)$:

$$-\int_0^L \lambda T''(x)v(x)dx = \int_0^L f(x)v(x)dx. \quad (3)$$

Next, using integration by parts, the equation becomes

$$-\int_0^L \lambda T''(x)v(x)dx = -[\lambda T'v]_{x=0}^{x=L} + \int_0^L \lambda T'(x)v'(x)dx. \quad (4)$$

If the test function v is required to satisfy the same Dirichlet boundary condition as the temperature T ,

$$v(x = 0) = 0, \quad (5)$$

and with the aid of the homogeneous Neumann boundary condition from Equation (2)

$$T'(x = L) = 0, \quad (6)$$

one can simplify Equation (4) as follows:

$$-\int_0^L \lambda T''(x)v(x)dx = \int_0^L \lambda T'(x)v'(x)dx. \quad (7)$$

After substitution, we arrive at the following equation

$$\int_0^L \lambda T'(x)v'(x)dx = \int_0^L f(x)v(x)dx. \quad (8)$$

The transformations used so far are equivalent, and no approximation has been made yet. The first transformation that involves approximation is the Galerkin projection introduced in the next step. The idea behind the Galerkin projection is that solving Equation (8) on a computer requires the infinite-dimensional space V to be replaced by a finite-dimensional subspace S . The problem in Equation (8) then becomes finding $T_S \in S$, so that

$$\int_0^L \lambda T'_S(x)v'(x)dx = \int_0^L f(x)v(x)dx \quad \forall v \in S. \quad (9)$$

If we take an arbitrary basis $\{\phi_i : 1 \leq i \leq n\}$, an element $T_S \in S$ can be represented as

$$T_S = \sum_{i=1}^n T_i \phi_i. \quad (10)$$

The procedure is visualized in Figure 2, where T (solid blue line) is approximated by T_S (dashed orange line) as a linear combination defined in Equation (9).

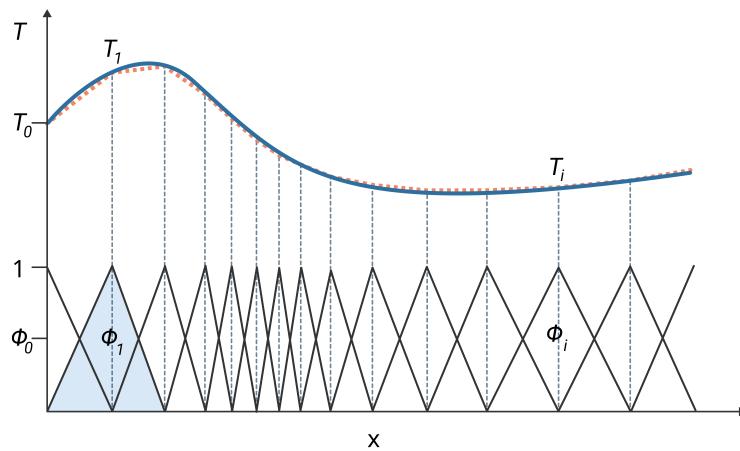


Figure 2. Visualization of T (solid blue line) being approximated by T_S (dashed orange line) as a linear combination of linear basis functions ϕ_i (solid black lines). A simple choice of basis functions would be the space of piecewise-linear functions on $[0, 1]$. Any piecewise-linear function can be uniquely expressed as a sum of scaled tents.

The substitution of the basis functions ϕ_i into both sides of Equation (9) results in the so-called weak formulation

$$\sum_{i=1}^n T_i \int_0^L \phi'_i(x) v'(x) dx = \frac{1}{\lambda} \int_0^L f(x) v(x) dx. \quad (11)$$

In Equation (11), the condition needs to be satisfied for all $v \in S$. Since v can be expressed as a linear combination of basis functions

$$v = \sum_{j=1}^n \beta_j \phi_j, \quad (12)$$

we can transform Equation (11) as follows:

$$\sum_{i=1}^n T_i \int_0^L \phi'_i(x) \phi'_j(x) dx = \frac{1}{\lambda} \int_0^L f(x) \phi_j(x) dx \quad \forall j \in \{1, \dots, n\}. \quad (13)$$

The system of linear equations can then be written as

$$A \tilde{T} = b, \quad (14)$$

where A is an $N \times N$ -dimensional symmetric and invertible matrix called the stiffness matrix, and \tilde{T} and b are N -dimensional vectors defined as

$$\begin{aligned} \tilde{T} &:= (T_i)_{i=1, \dots, n} \\ A &:= \left(\int_0^L \phi'_i(x) \phi'_j(x) dx \right)_{i,j=1, \dots, n} \\ b &:= \left(\frac{1}{\lambda} \int_0^L f(x) \phi_j(x) dx \right)_{j=1, \dots, n} \end{aligned} \quad (15)$$

Usually, it is advisable to use preconditioners on the system matrix before solving it for a better convergence time, which is covered by [40–42]. The usage of preconditioners is especially important, in most cases, in CFD simulations.

3.2. Quantum Solver for System of Linear Equations

Various quantum algorithms for FEM have been proposed lately that claim to be exponentially faster than their classical counterparts, but it is hard to decide which method has the most potential as of now, especially with new algorithms being developed each year. Most of these algorithms utilize the famous Harrow–Hassidim–Lloyd (HHL) algorithm [32], which solves a system of linear equations. In this paper, we will treat it as a quantum black box, i.e., we are solely interested in the output returned by this quantum subroutine.

The HHL algorithm prepares the state $|x'\rangle = \frac{|x'\rangle}{\|x'\|}$ that encodes the solution of the equation

$$A |x'\rangle = |b\rangle. \quad (16)$$

Let $\|\cdot\|$ be a norm (usually taken as $\|\cdot\|_2$) and A be an $N \times N$ Hermitian matrix such that $\|A\| \cdot \|A^{-1}\| \leq k$ and A has, at most, s nonzero entries in each row. Assume that there is an algorithm \mathcal{P}_A that, on input (r, i) , outputs the location and value of the i th nonzero entry in row r . Let b be an N -dimensional unit vector, and assume that there is an algorithm \mathcal{P}_b that produces the corresponding state $|b\rangle$. Since A is a Hermitian matrix, it has a spectral decomposition:

$$A = \sum_{j=0}^{N-1} \lambda_j |u_j\rangle \langle u_j|, \quad \lambda_j \in \mathbb{R}, \quad (17)$$

where (u_j, λ_j) is the j th eigenvector, eigenvalue pair of A . In the eigenbasis, A is diagonal; therefore,

$$A^{-1} = \sum_{j=0}^{N-1} \lambda_j^{-1} |u_j\rangle \langle u_j|. \quad (18)$$

The right-hand side of Equation (16) can also be rewritten in the eigenbasis as

$$|b\rangle = \sum_{j=0}^{N-1} b_j |u_j\rangle, \quad b_j \in \mathbb{C}. \quad (19)$$

The HHL algorithm solves the following equivalent equations

$$|x'\rangle = A^{-1} |b\rangle = \sum_{j=0}^{N-1} \lambda_j^{-1} b_j |u_j\rangle, \quad |x\rangle = \frac{|x'\rangle}{\|x'\|}, \quad (20)$$

by producing the state $|x\rangle$ up to accuracy ϵ in l_2 norm, with a bounded probability of failure, and makes

$$O(sk \text{polylog}(sk/\epsilon))$$

uses of \mathcal{P}_A and \mathcal{P}_b , and the run time is the same up to a $\text{polylog}(N)$ factor (where the $\text{polylog}(x)$ denotes the polynomial in terms of $\log(x)$).

As mentioned in [28,29], if one hopes to preserve the quantum speedup, two operations are not allowed. The first one is the input state preparation $|b\rangle$ with an encoded quantum circuit, where reading all data entries requires $\mathcal{O}(N)$ time. The second one is measuring the output state to extract the state to a classical vector, which also requires $\mathcal{O}(N)$ time. Solving a time-dependent problem iteratively requires the discretized physical governing equation to be solved at every time step; thus, the two problems mentioned above exist in every time step. An additional $\mathcal{O}(N)$ time complexity cannot be allowed in the calculations if an exponential speedup is to be preserved.

One of the workarounds for this problem is using QRAM as mentioned in [29,43]. Following the work of [29], the runtime analysis concludes that the time complexity of their quantum procedure is

$$\mathcal{O}\left(\frac{(s^3 + \log N)sk \log^3 N}{\epsilon^2} \text{polylog}(sk/\epsilon)\right).$$

The classical counterpart's time complexity is

$$\mathcal{O}(Nsk \log 1/\epsilon)$$

when using the conjugate gradient (CG) as the linear solver. The accuracy parameter ϵ has also been incorporated in their quantum time complexity, which is an important factor that has been left out in a few previous attempts claiming to reach exponential speedup incorrectly, as pointed out by Montanaro [28]. The algorithm they developed outperforms the classical algorithm when the relation $N \gg 1/\epsilon^2$ holds.

3.3. Reducing Model Complexity

Problems in CFD often lead to large systems of equations that are computationally too expensive to solve on classical computers. Model order reduction methods aim to solve this problem by decreasing the complexity and computational cost of a system by removing “redundant” information from the high fidelity model. The main idea is to generate computationally efficient ROMs that work with a relatively high accuracy at points of interest and to remove any unnecessary complications in the model otherwise. One example is the proper orthogonal decomposition (POD) method, which is a widely

used snapshot-based reduction method for CFD applications. The details of the POD method are discussed next in Section 3.3.1.

Snapshot-based model order reduction methods require a set of solutions that are computed or measured in advance. Standard MOR techniques assume that the snapshots, which are solutions sampled at different and evenly spaced times, use a constant spatial mesh. Adaptive snapshot computation methods also exist, where the mesh could be different in the case of each snapshot [44]. The solution of the ROM is represented as a linear combination of the snapshots, with the coefficients determined by a Galerkin projection based on a weak formulation of the governing equations. The two most commonly used MOR methods for nonlinear systems are the proper orthogonal decomposition (POD) [45–47] and trajectory piecewise-linear techniques (TPWL) [48–51], which are discussed in Section 3.3.1 and Section 3.3.2, respectively.

3.3.1. Proper Orthogonal Decomposition

The POD method generates a set of reduced basis functions from the solutions that are computed or measured in advance. The goal is to obtain an optimal low-dimensional basis for representing high-dimensional experimental or simulation data. This low-dimensional basis can be used to formulate ROMs of models with a high complexity. The main assumption in the snapshot-based MOR techniques is that an approximation to the solution of the problem can be expressed as a linear combination of spatial modes. Let $u(x, t)$ denote a given flow field and $u_r(x, t)$ denote its approximation in the reduced space. The POD method decomposes $u(x, t)$ into an orthonormal system of spatial modes $\varphi_i(x)$ with temporal coefficients $a_i(t)$ as follows:

$$u(x, t) \approx u_r(x, t) = \sum_{i=1}^{N_r} a_i(t) \varphi_i(x). \quad (21)$$

The spatial modes have to be carefully selected, as a correct choice of basis functions could lead to an efficient ROM with a reduced simulation time and high accuracy in the online simulation phase. The POD is capable of selecting the spatial modes with the highest energies; thus, the method is usually a good choice for fluid flow applications. The POD basis

$$X_{N_r}^{POD} = \text{span}\{\varphi_i\}, \quad i = 1, \dots, N_r, \quad (22)$$

can be built from a set of snapshots of velocity solutions

$$u_n(x) = u(x, t_n), \quad n = 1, \dots, N_S, \quad (23)$$

where N_r denotes the number of ROM functions and N_S denotes the number of snapshots taken from the obtained solutions. By definition, the POD basis calculation can be thought of as the least square method, which minimizes the difference between the snapshots and the projection of the snapshots on the spatial modes in the L^2 -norm. If the L^2 -norm is chosen, the POD basis is optimal, considering the energy contained in the snapshots:

$$X_{N_r}^{POD} = \arg \min \frac{1}{N_S} \sum_{n=1}^{N_S} \|u_n(x) - \sum_{i=1}^{N_r} \langle u_n(x), \varphi_i(x) \rangle_{L^2} \varphi_i(x)\|_{L^2}^2, \quad (24)$$

$$\langle \varphi_i(x), \varphi_j(x) \rangle_{L^2} = \delta_{i,j}.$$

In order to solve Equation (24), the following eigenvalue problem is considered:

$$C\xi = \lambda_i \xi, \quad i = 1, \dots, N_S, \quad (25)$$

where $C \in \mathbb{R}^{N_S \times N_S}$ is the correlation matrix, whose components are calculated as follows:

$$[C]_{kl} = \frac{1}{N_r} \langle u_k(x), u_l(x) \rangle_{L^2}. \quad (26)$$

The (λ_i, ξ_i) eigenvalue–eigenvector pair is used to construct the POD basis functions

$$\varphi_i(x) = \frac{1}{\sqrt{\lambda_i}} \sum_{n=1}^{N_s} \xi_{i,n} u_n(x), \quad i = 1, \dots, N_r. \quad (27)$$

Since the eigenvalues are sorted into descending order, the first modes have the property of retaining most of the energy present in the original solutions. The obtained basis functions are orthogonal and can be normalized in order to obtain

$$\langle \varphi_i(x), \varphi_j(x) \rangle_{L^2} = \delta_{ij}, \quad (28)$$

where δ_{ij} is the Kronecker delta function. Replacing the velocity u with u_r in the incompressible Navier–Stokes equations, as seen in [46], and applying a Galerkin projection on the POD basis functions, the following POD-Galerkin ROM is obtained:

$$\frac{da_j(t)}{dt} = v \sum_{i=1}^{N_r} B_{ji} a_i(t) - \sum_{k=1}^{N_r} \sum_{i=1}^{N_r} C_{jki} a_k(t) a_i(t), \quad j = 1, \dots, N_r, \quad (29)$$

where

$$\begin{aligned} B_{ji} &= \langle \nabla \varphi_j, \nabla \varphi_i \rangle_{L^2} \\ C_{jki} &= \langle \varphi_j, (\varphi_k \cdot \nabla) \varphi_i \rangle_{L^2} \\ a_j(0) &= \langle \varphi_j, u_1(x) \rangle_{L^2}. \end{aligned} \quad (30)$$

One can obtain the ROM by a projection of the state vector x using the POD basis functions

$$x(t) = V \hat{x}(t), \quad (31)$$

where $\hat{x}(t)$ is an m -dimensional vector in the reduced space, containing the time-dependent amplitudes of m basis vectors, contained in the columns of the projection matrix V . After applying the projection to the nonlinear system, the resulting ROM is of the form

$$\begin{aligned} \dot{\hat{x}}(t) &= V^T f(V \hat{x}(t), u(t)) \\ \dot{y}(t) &= h(V \hat{x}(t)). \end{aligned} \quad (32)$$

As for the time-domain discretization, a second-order backward Euler time discretization scheme is used in order to preserve the stability of the discretized system:

$$\dot{\hat{x}}(t) \approx \frac{3\hat{x}^n - 4\hat{x}^{n-1} + \hat{x}^{n-2}}{2\Delta t} = V^T f(V \hat{x}^n, u^n), \quad (33)$$

where Δt denotes the time step and the superscript n denotes the time index of the current solution. Subtracting the right side from Equation (33), the nonlinear system of equations simplifies to

$$F(\hat{x}^n) = 3\hat{x}^n - 4\hat{x}^{n-1} + \hat{x}^{n-2} - 2\Delta t V^T f(V \hat{x}^n, u^n) = 0, \quad (34)$$

for \hat{x}^n , which is typically performed using Newton iteration. It is important to note that the Jacobian needs to be evaluated at every iteration at a given time step to obtain the flux term $f(\cdot)$. This implies that the simulation of ROM could be no more efficient than the original system, since the flux term evaluation takes most of the computation time. This issue will be addressed in the next section. For an error assessment on the accuracy of the POD reduction method, refer to [44].

3.3.2. Reduced-Order Linearized Model

The motivation behind trajectory piecewise-linear techniques is to linearize nonlinear systems around suitably selected states and to approximate the system by combining a set of linearized reduced-order models via a weighting procedure. Let $(x_0, u_0), \dots, (x_{S-1}, u_{S-1})$ be states obtained from previous simulations of the FOM on some finite time interval $[t_{start}, t_{end}]$. Efficient linearized models can be extracted from the system by using a Taylor expansion of the nonlinearity around the states (x_i, u_i)

$$\begin{aligned} f(x, u) = & f(x_i, u_i) + A_i(x - x_i) + B_i(u - u_i) \\ & + \frac{1}{2} W_i(x - x_i) \otimes (x - x_i) + \dots, \end{aligned} \quad (35)$$

where \otimes is the Kronecker product and A_i and W_i are the Jacobian and the Hessian of $f(\cdot)$ evaluated at the state (x_i, u_i) . The matrix $B_i = \frac{\partial f}{\partial u}$ is also evaluated at (x_i, u_i) . By removing the quadratic and higher terms, the nonlinear system can be linearized around any given state to yield a state-space model of the form:

$$\begin{aligned} \dot{x}(t) = & A_i x(t) + B_i u(t) + [f(x_i, u_i) - A_i x_i(t) - B_i u_i] \\ y(t) = & C_i x(t), \end{aligned} \quad (36)$$

where $C_i = \frac{\partial h}{\partial x}$ is also evaluated at (x_i, u_i) . The vector of unknowns $x(t)$ can be written as

$$x(t) = x_i + x'_i(t), \quad (37)$$

where x_i , fixed in time, is the value of state vector x at the linearization point i , and $x'_i(t)$ contains the perturbation of the n unknown flow quantities about that linearization point x_i . The linearized equation can then be expressed as

$$\begin{aligned} \dot{x}'_i(t) = & A_i x'_i(t) + B_{1i} u(t) + B_{2i} \\ y(t) = & C_i x'_i(t) + C_{0i}, \end{aligned} \quad (38)$$

where $B_{2i} = f(x_i, u_i) - B_i u_i$ and $C_{0i} = C_i x_i$. Since each of the linearizations approximates the nonlinear system around the expansion point x_i , a model including all of these linearizations could approximate the original system over a larger time interval and a larger part of the state-space.

Plotted in Figure 3 are four linearization points, x_0, x_1, x_2 and x_3 , along a “training trajectory” A, which is obtained through a simulation of the original nonlinear system. The goal is to capture the most relevant dynamics of the system through a series of simulations in the offline training phase with inputs that reflect the dynamics of interest for the given application. Each linearized model has its own range of validity (denoted by the blue circles) specifying the range where approximation by linearization can be considered as accurate. Exploiting quantum algorithms for their exponential speedup in the offline training part could prove to be the most beneficial, as it is a computationally expensive part of the TPWL.

Using piecewise-linearized models for simulations might already be faster than simulating the original system. However, the linearized models can be further reduced by using previously described MOR techniques. One additional advantage of linearizing the system first is that it allows for the possibility of applying both linear and nonlinear MOR methods to the linear submodels. As a few examples for linear MOR methods, in [48], Rewiénksi proposes the usage of Krylov-based reduction using the Arnoldi method. In [52] Vasilyev, Rewiénksi and White introduce balanced truncation to TPWL, and Voß [53] uses Poor Man’s TBR as a linear MOR kernel.

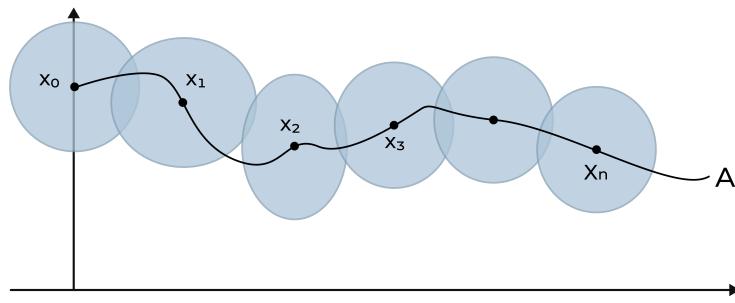


Figure 3. Collection of linearization points x_0, x_1, x_2 and x_3 in a 2D state-space. Blue circles denote suitable regions for use at each linearization point. Trajectory A is called the training trajectory.

In this paper, we will continue the work described in Section 3.3.1 and use proper orthogonal decomposition (POD) for the reduction step. A reduced-order linearized model can be obtained by applying the projection defined before in Equation (31) to the system in Equation (38), yielding

$$\begin{aligned} \frac{\partial}{\partial t} \hat{x}'_i(t) &= \hat{A}_i \hat{x}'_i(t) + \hat{B}_{1i} u(t) + \hat{B}_{2i} \\ \hat{y}_i(t) &= \hat{C}_i \hat{x}'_i(t) + C_{0i}, \end{aligned} \quad (39)$$

where the reduced-order matrices are given by

$$\begin{aligned} \hat{A}_i &= V^T A_i V \\ \hat{B}_{1i} &= V^T B_{1i} \\ \hat{B}_{2i} &= V^T B_{2i} \\ \hat{C}_i &= C_i V. \end{aligned} \quad (40)$$

Each ROM has its own range of validity, as seen in the updated Figure 4 denoted by the orange circles. The orange dashed line represents the approximation made to the training trajectory A calculated along the points x'_1, x'_2 and x'_3 in the reduced space, which correspond to x_1, x_2 and x_3 points in the original system.

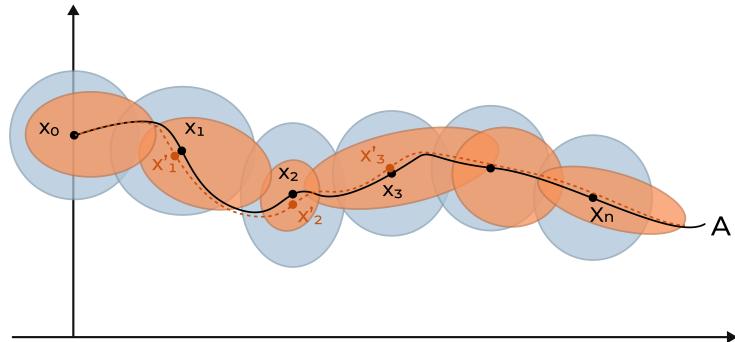


Figure 4. Collection of reduction points x'_1, x'_2 and x'_3 , along with the previously highlighted linearization points x_0, x_1, x_2 and x_3 in a 2D state-space. Orange circles denote suitable regions for use for each reduced-order model, whereas the orange dashed line represents an approximation to the training trajectory A.

The last step is to apply a weighting function on the ROMs. Using the procedure described in [48], the piecewise-linear ROM becomes

$$\sum_{i=0}^{s-1} w_i(x) \left\{ \frac{\partial}{\partial t} \hat{x}'_i(t) = \hat{A}_i \hat{x}'_i(t) + \hat{B}_{1i} u(t) + \hat{B}_{2i} \right\}, \quad (41)$$

$$\sum_{i=0}^{s-1} w_i(x) \left\{ \hat{y}_i(t) = \hat{C}_i \hat{x}'_i(t) + C_{0i} \right\},$$

where $w_i(x)$ are state-dependent weights. A typical choice is to let $w_i(x)$ be large for $x = x(t)$ close to x_i , and small otherwise. For other, more advanced schemes, refer to [50].

4. Quantum Reduced Finite Element Method

In this section, we will describe the different steps and details of our proposed method, combining all the preliminaries that we have seen in the previous sections. Our main motivation was to develop an approach where challenging problems in CFD applications can be accurately simulated with a minimization in computational time and development costs. The approach we have found combines quantum and classical algorithms that output a reusable set of linearized reduced-order models that can be efficiently simulated with different excitation sets on classical computers.

The steps in our method are briefly summarized in Figure 5, starting with the modeling of the physically relevant system that we plan to simulate numerically. This can be carried out in many commercially available FEA softwares, such as COMSOL, Abaqus, Ansys, OpenFOAM and Siemens NX, just to name a few of the most widely used ones. After obtaining an accurate model of the physical system, along with the governing equations and properties, we have to solve the originally large system for some initial states, which are called trajectories (this step is also called the offline training). The initial trajectories have to be carefully selected in order to obtain accurate results for the different excitation sets that we would like to simulate the physical system with using the reduced state later on.

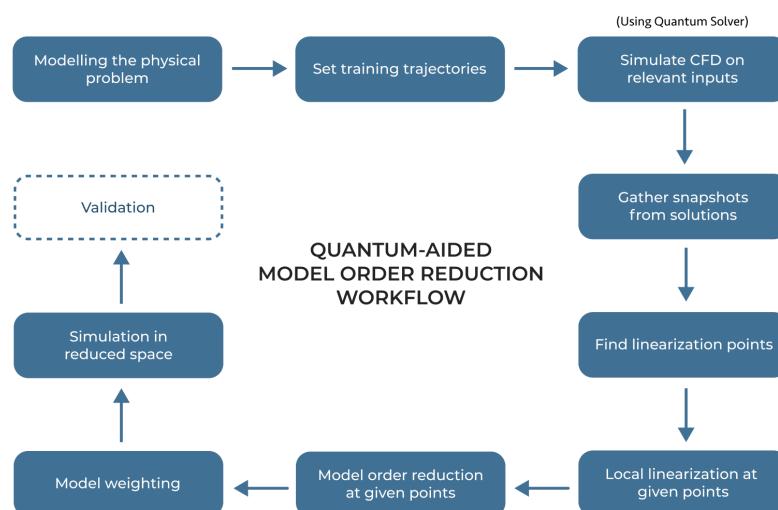


Figure 5. Visualization of the proposed quantum-aided model order reduction method highlighting the different steps in the workflow. This method can be used to generate linearized reduced-order models for efficiently solving complex simulations approximately with a high accuracy on classical computers. It is a combination of quantum and classical algorithms, where a quantum algorithm is used for solving the originally large problem in the offline training phase, with the results being post-processed by classical MOR techniques.

The quantum solver in this context is taken as a black box that is assumed to solve the system of linear equations (exponentially) faster than existing classical algorithms in every step of the time integration. This is the part where we can achieve the most speedup, but it can also be the most expensive step if we are not careful enough and use the quantum hardware for more time than necessary, assuming that the price of a QCPU is much higher

than that of a regular CPU. The solutions that we obtain from the solver need to be post-processed classically in order to acquire a reusable system of linearized reduced-order models, as seen in Section 3.3. For this, we first need to find the critical points in our system that hold the most nonlinearity and to linearize the system around these points. After the local linearization, we can start the reduction procedure at the given points to gather a set of reduced linear models.

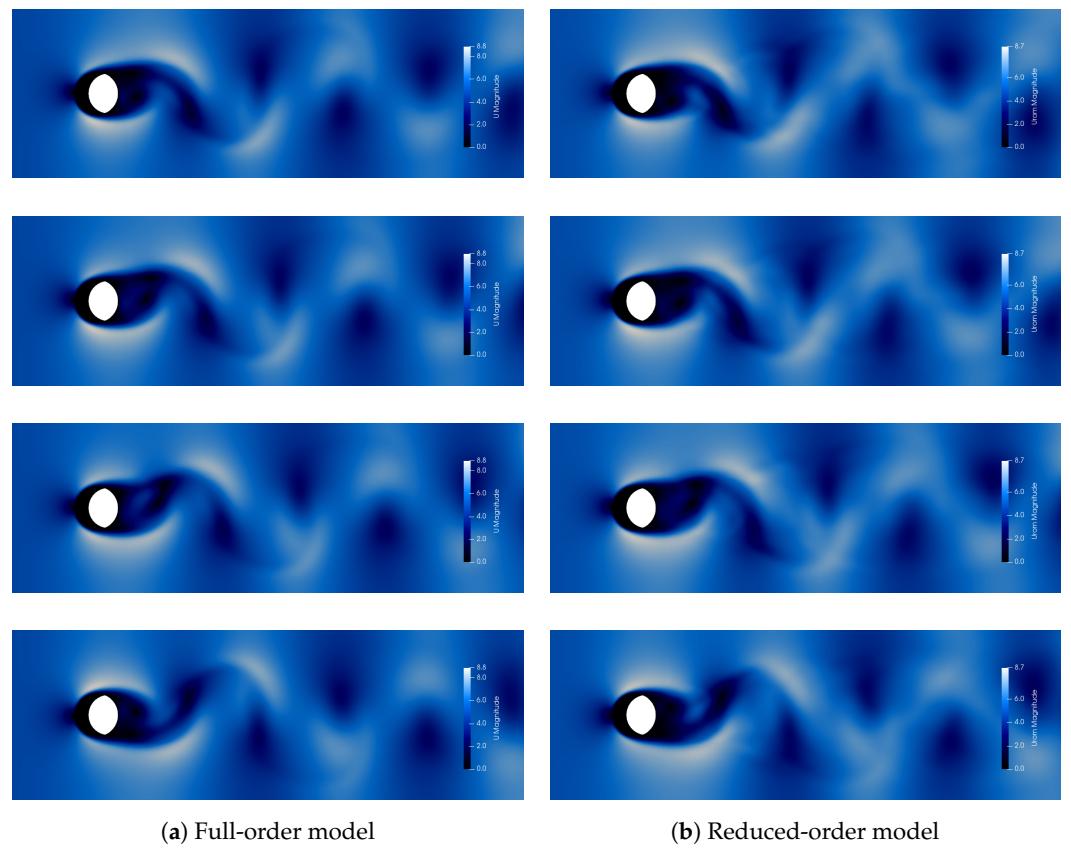
In the last step, a weight has to be assigned to each model. The weighting function typically assigns a large weight to models close to the given point in the simulation and assigns a small weight otherwise, as seen in Equation (41). The result is a weighted linearized set of ROMs that can be used for approximating the solutions of the original system. This approximation can be carried out with different parameter sets, besides what was used at the offline training part (but within its range of validity) on classical computers with much efficiency. The disadvantage of the TPWL is that it is only applicable for excitations that are in the validation range of the offline training part. If different excitations need to be used that are outside the validation range, a new offline training part needs to be carried out, and the quantum solver has to be applied again in that case.

Additional steps could include an evaluation of the results and validation by comparing the gathered solutions to the ones obtained by solving the original full system. As there are no general rules or prior error analysis that we can make for estimating the accuracy of ROMs, experimenting with different configurations might be necessary initially when the models are yet to be fine-tuned for a specific problem in order to achieve more accurate results or a better performance.

4.1. Study: Flow Past a Cylinder

As a demonstration of the POD reduction method, we show an example of an unsteady, incompressible flow past a cylinder placed in a channel at a right angle to the oncoming fluid. The cylinder is offset from the center of the flow to make the steady-state symmetrical flow unstable. Vortex shedding is an interesting and well-known phenomenon in fluid dynamics that has been studied in relation with MOR techniques for reduction purposes [47,54,55]. For Reynolds numbers below 100, the flow can be treated as a steady laminar flow, and for values above 100, the flow starts developing turbulent behavior. In this simulation, the Reynolds number is equal to 161, which is already enough to show signs of the Karman vortex street, as seen in Figure 6.

The geometry definition and the simulation were carried out using OpenFOAM 2012 following the work of Kornbleuth [56]. The full-order model has 85,720 degrees of freedom. The simulation was run for 3 s, with a fixed time step of $dt = 0.00001$ on eight cores in parallel on an Intel(R) Xeon(R) Gold 6248R, 3.00 GHz CPU for a total time of 10.309 s (≈ 171 min). For the model-order reduction step, Illinois Rocstar LLC's AccelerateCFD was used, which is an open-source software that reduces a full-order CFD flow (defined in OpenFOAM) using the global POD reduction method. The POD basis was calculated using a total number of 250 snapshots evenly collected from the solutions of OpenFOAM. In Figure 7, we can see the cumulative flow energy in terms of how many basis vectors are selected. This can be used to decide the number of POD basis vectors used for computing the ROM.



(a) Full-order model

(b) Reduced-order model

Figure 6. Incompressible flow past a cylinder placed in a channel at a right angle to the oncoming fluid. The simulation was carried out in OpenFOAM, with Reynolds number set to 161. Comparison of velocity U between the full-order model (a) and the reduced-order model solutions (b).

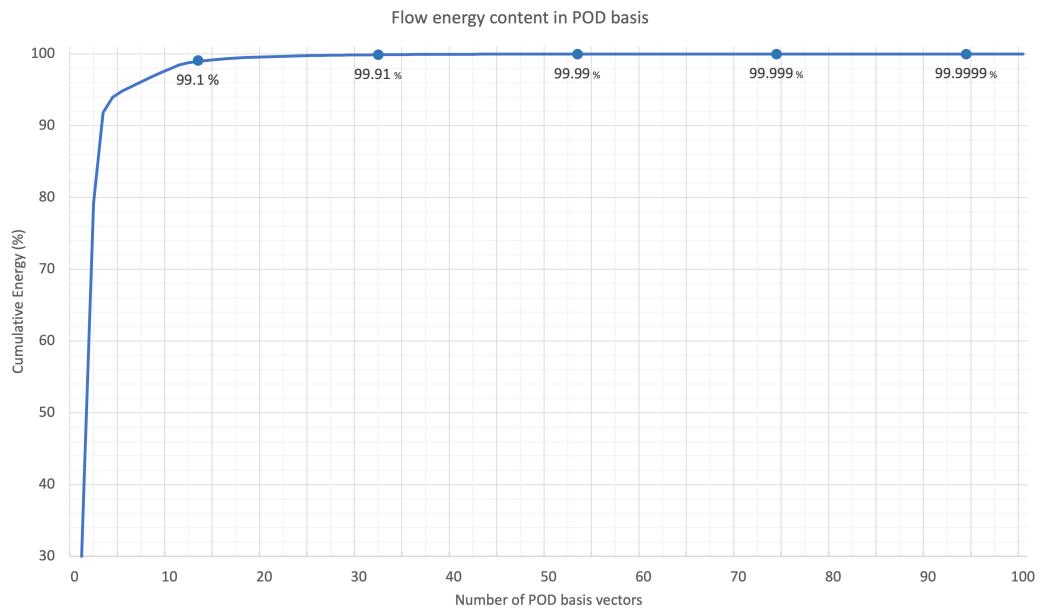


Figure 7. Cumulative energies of the obtained POD base vectors. The first 95 POD bases vectors have a cumulative energy above 99.9%, which means that the rest of the base vectors can be dismissed while still maintaining a high accuracy.

The ROM simulations were carried out using different values of the POD velocity space dimension $N_r = 14, 33, 54, 75, 95$. In these cases, the cumulative energies are 99.1%,

99.91%, 99.99%, 99.999% and 99.9999%, respectively. The simulations were run in series on the same processor that was used for the FOM simulation. The ROM simulation results can be seen in Table 1.

Naturally, the proposed method is optimal in terms of computational time and expenses compared to using either classical or quantum solvers alone. Since, for the offline training phase, a quantum solver is used, in this phase, the computations could be exponentially faster compared to a classical solver. On the other hand, if we consider the online simulation phase after the model-order reduction, the system's small complexity allows for a much faster simulation time compared to both quantum and classical solvers, as the reduction and speedup factor can be substantial, as seen in Table 1. In this paper, $Re = 161$ was used for simplicity, but for simulations of flow past a cylinder at high Reynolds numbers, refer to [57,58]. For POD reduction methods applied to turbulent flows with high Reynolds numbers, refer to [59–61].

Table 1. Comparison of ROMs calculated with different values of the POD velocity space dimension. The higher the number of POD basis functions is, the more accurate the results are, but with a decrease in the speedup factor.

Comparison of ROMs					
Number of POD basis functions:	14	33	54	75	95
Cumulative energy:	99.1%	99.91%	99.99%	99.999%	99.9999%
Pre-processing:	43.55s	59.24s	87.81s	160.09s	286.38s
Computation:	1.67s	11.09s	44.7s	118.52s	245.52s
Post-processing	18.35s	20.18s	21.31s	22.37s	23.8s
Overall time:	63.57s	90.51s	153.82s	300.98s	555.7s
Speedup factor:	1297x	911x	536x	274x	148x

4.2. Additional Quantum Speedup

As mentioned before, measuring the quantum states and preparing the input at each time step requires $\mathcal{O}(N)$ time. This $\mathcal{O}(N)$ multiplier can cancel out the speedup gained from using the quantum solver; therefore, if one aims for an exponential speedup, this problem needs to be addressed. However, with the use of snapshot-based MOR methods, it is possible to slightly speed up the computational time in favor of the quantum algorithm. The reason is that, in snapshot-based reduction methods, such as the POD described in Section 3.3.1, knowing the solution is not required at every time step. Rather, we need solutions sampled at different and evenly spaced times. This means that measuring the quantum state is only required at each snapshot. Since the described MOR methods do not require any addition preprocessing steps for the measured and prepared solutions, the $\mathcal{O}(N)$ time complexity remains unaffected. Nevertheless, finding the sufficient number of snapshots depends on the desired accuracy, and can vary greatly between applications. This makes it difficult to estimate what the optimal number of snapshots is in general, and requires a deep analysis and careful experimenting of the problem at hand. However, it is safe to say that, in the case where we select only every k th time step as a snapshot, then one can reduce the $\mathcal{O}(N)$ time complexity caused by measuring the quantum state at each time step by a factor of k . If every time step is selected as a snapshot, then there would be no additional speedup, and we would arrive back at the original time complexity.

Throughout this paper, we assumed that the snapshots are evenly spaced in the time domain, but there are more advanced adaptive methods for the snapshot selection [62–64]. With the use of such methods, the additional speedup could be even more significant.

5. Conclusions

In this paper, we have proposed a new method for solving computationally complex problems in CFD applications with a substantial speedup and with a reduced development cost compared to standalone classical or quantum solvers. We have achieved this by combining quantum algorithms for solving systems of linear equations with suitable classical algorithms for data processing and reducing the complexity of the given problem (TPWL-POD). We carry this out by discretizing the physical model using FEM and solving the resulting system of linear equations for an initial training set of inputs. In the training stage, a suitable quantum algorithm is applied for solving the system of equations with an exponential speedup. In the second part, we use snapshot-based classical model order reduction techniques to reduce the complexity of the original problem, making it eligible to be solved classically with different excitations in future simulations, avoiding the need for a quantum solver. This set of linearized reduced-order models can be used for real-time calculations in embedded systems, model validation and optimization purposes.

It is important to note that it is only possible to obtain accurate solutions if the parameter sets are in the range of the validity of the training set. Since it can be difficult to define the training trajectory in an optimal way, covering all the dynamics of interest of the system beforehand, this might require some experimenting at first. The positive side is that the training step only needs to be carried out once before the reduced models could be used for simulations, but any changes in the model, geometry and time-domain would require the retraining of the models. This would increase the number of times that the quantum solver is used, along with the development cost.

After obtaining a set of linearized reduced-order models, they can be used for simulating the physical system with different excitation sets. Since the complexity and size of the original problem is greatly reduced at this point, it is clearly faster to solve the reduced models classically than to solve the original problem. We would also argue that solving the reduced models could be computationally faster on a classical computer than solving the initial problem with a quantum solver. If the system is large enough and has to be simulated with many different excitations or is used in embedded systems for real-time calculations, our hybrid approach could mean a faster computational time compared to both classical and quantum algorithms, and since a mostly reduced version of the original problem is solved on classical CPUs, the cost would be much less than a full quantum or classical solution each time with the full sized matrix.

One of the other main advantages of our proposed method is that we only require the existence of the solutions, and do not have any prerequisites on how they are produced, meaning that it is general and flexible enough to replace the quantum solver and the reduction method for a more suitable one for a given application. This also makes it hard to estimate the limitations and constraints of our method, as it highly depends on the used algorithms. Therefore, one of the disadvantages of this method would be the difficulties of automatization, as there are no general rules for defining the training trajectories and critical points, as it can vary with each application.

6. Future Works

Regarding the next steps, we would like to scale up our approach to other applications besides CFD. This would include both linear and nonlinear use cases, such as thermodynamics, structural mechanics, electromagnetics and coupled multiphysics models, just to name a few. This would allow for the generalization of our method in order to use it for a wide range of industrial problems in the field of numerical simulations.

The majority of quantum solvers promise exponential speedup compared to classical algorithms for problems where the problem domain is large enough. Thus, we can make our method more efficient by making a prior estimation of whether it is actually worth using a quantum solver as opposed to a classical one in terms of the computational time depending on the problem's complexity. A similar estimation could also be given for the reduced models, as their size could still be large enough to gain quantum speedup. This

estimation is visualized as a decision tree in Figure 8. If the size is close to the boundaries of gaining any speedup from the quantum algorithm, an additional trade-off could be made. If the size of the ROM is further reduced, it can be solved faster and cheaper classically, but with less accuracy (as when we reduced the model size, we lost additional information). On the other hand, if the size of the ROM is kept at a larger size, it can be solved faster and more accurately with a quantum algorithm, but with a higher cost.

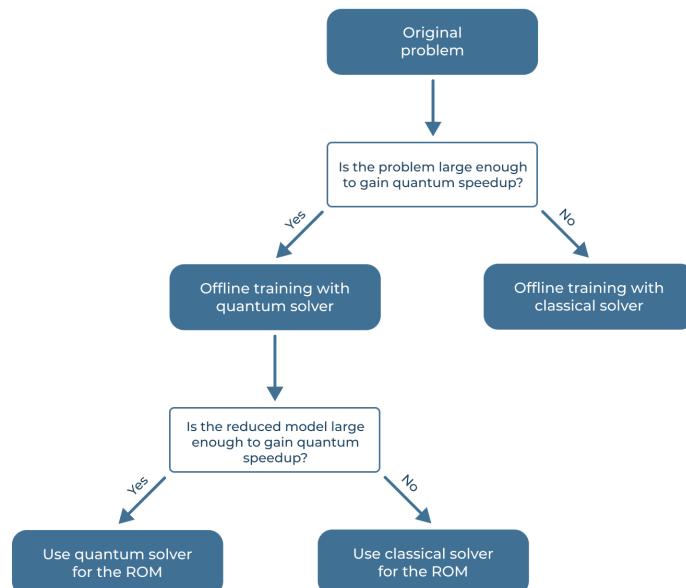


Figure 8. Decision tree for whether it is worth using quantum solver over classical ones. The same question can be applied for both the original large system and its corresponding reduced model.

Simulation models are often parameterized to allow for different variations in the geometry, shape, material, loading, boundary and initial conditions during their design. The described MOR methods in this paper are not robust enough with respect to parametric variations, and a new set of ROMs needs to be generated each time a parameter is changed in the system. This limits the applicability of our method for product design and optimization problems, as the whole workflow needs to be reset and started from scratch. Thus, another important work would be the development and integration of parametric reduced-order models (PROMs) into our workflow, which could reflect the parameter-dependent behavior in the reduced-order systems. Parametric model order reduction (PMOR) methods are intended to solve this exact problem, and could benefit greatly in product design and optimization problems, where exploration cycles can be achieved by carrying out the parametric simulations in the reduced space [65–67] and with parametrized geometries [68,69].

Author Contributions: Conceptualization, S.J., Z.Z., A.K., T.M.; methodology, S.J., Z.Z., A.K., T.M.; software, S.J.; validation, S.J., Z.Z., A.K., T.M.; investigation, S.J., Z.Z., A.K., T.M.; writing—original draft preparation, S.J., Z.Z., A.K., T.M.; writing—review and editing, S.J., Z.Z., A.K., T.M.; supervision, A.K., Z.Z.; project administration, A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by grants of the “Application Domain Specific Highly Reliable IT Solutions” project, which has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Subprogramme) funding scheme. This research was also supported by the Ministry of Innovation and Technology and the National Research, Development and Innovation Office within the Quantum Information National Laboratory of Hungary (NKFIH-873-1/2020).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FE	finite element
FEA	finite element analysis
FEM	finite element method
FVM	finite volume method
FDM	finite difference method
FOM	full-order model
MOR	model order reduction
ROM	reduced-order model
PROM	parametrized reduced-order model
HPC	high performance computing
QRAM	quantum random access memory
HHL	Harrow–Hassidim–Lloyd
CFD	computational fluid dynamics
BVP	boundary value problem
QLE	quantum linear equation
CG	conjugate gradient
POD	proper orthogonal decomposition
PDE	partial differential equation
TPWL	trajectory piecewise-linear
NSE	Navier–Stokes equations
t	time (seconds)
T	temperature (Celsius)
λ	thermal conductivity (W/(cm · C))
U	velocity (m/s)

References

1. Sato, K.; Yoshitake, A.; Hosoya, Y.; Mikami, H. *FEM Simulation to Estimate Crashworthiness of Automotive Parts* (No. 982356); SAE Technical Paper; International Body Engineering Conference & Exposition: Detroit, MI, USA, 1998.
2. Venkatesh, J.; Murthy, M.P. Design and structural analysis of high speed helical gear using Ansys. *Int. J. Eng. Res. Appl.* **2014**, *2*, 215–232.
3. Johnson, F.T.; Tinoco, E.N.; Yu, N.J. Thirty years of development and application of CFD at Boeing Commercial Airplanes, Seattle. *Comput. Fluids* **2005**, *34*, 1115–1151. [[CrossRef](#)]
4. LeVeque, R.J. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*; Society for Industrial and Applied Mathematics: Seattle, DC, USA, 2007.
5. Hu, H.H.; Patankar, N.A.; Zhu, M.Y. Direct numerical simulations of fluid–solid systems using the arbitrary Lagrangian–Eulerian technique. *J. Comput. Phys.* **2001**, *169*, 427–462. [[CrossRef](#)]
6. Zienkiewicz, O.C.; Taylor, R.L.; Zhu, J.Z. *The Finite Element Method: Its Basis and Fundamentals*; Elsevier: Oxford, UK, 2005.
7. Axelsson, O.; Barker, V.A. *Finite Element Solution of Boundary Value Problems: Theory and Computation*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2001.
8. Rao, S.S. *The Finite Element Method in Engineering*; Butterworth-Heinemann: Boston, MA, USA, 2017.
9. Versteeg, H.K.; Malalasekera, W. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*; Pearson education: Harlow, UK, 2007.
10. Moukalled, F.; Mangani, L.; Darwish, M. The finite volume method. In *The Finite Volume Method in Computational Fluid Dynamics*; Springer: Cham, Switzerland, 2016; pp. 103–135.
11. Jasak, H. Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows. Ph.D. Thesis, University of London, London, UK, 1996.
12. Launder, B.E.; Spalding, D.B. The numerical computation of turbulent flows. In *Numerical Prediction of Flow, Heat Transfer, Turbulence and Combustion*; Pergamon: New York, NY, USA, 1983; pp. 96–116.
13. Launder, B.E.; Sandham, N.D. (Eds.) *Closure Strategies for Turbulent and Transitional Flows*; Cambridge University Press: Cambridge, UK, 2002.
14. Argyropoulos, C.D.; Markatos, N.C. Recent advances on the numerical modelling of turbulent flows. *Appl. Math. Model.* **2015**, *39*, 693–732. [[CrossRef](#)]
15. Antoulas, A.C.; Ionutiu, R.; Martins, N.; ter Maten, E.J.W.; Mohaghegh, K.; Pulch, R.; Striebel, M. Model order reduction: Methods, concepts and properties. In *Coupled Multiscale Simulation and Optimization in Nanoelectronics*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 159–265.

16. Schilders, W. Introduction to model order reduction. In *Model Order Reduction: Theory, Research Aspects and Applications*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 3–32.
17. Benner, P.; Schilders, W.; Grivet-Talocia, S.; Quarteroni, A.; Rozza, G.; Miguel Silveira, L. *System-and Data-Driven Methods and Algorithms*; De Gruyter: Berlin, Germany, 2021; p. 378.
18. Benner, P.; Schilders, W.; Grivet-Talocia, S.; Quarteroni, A.; Rozza, G.; Miguel Silveira, L. *Model Order Reduction: Volume 2: Snapshot-Based Methods and Algorithms*; De Gruyter: Berlin, Germany, 2020; p. 348.
19. Demo, N.; Ortali, G.; Gustin, G.; Rozza, G.; Lavini, G. An efficient computational framework for naval shape design and optimization problems by means of data-driven reduced order modeling techniques. *Boll. Dell'Unione Mat. Ital.* **2021**, *14*, 211–230. [[CrossRef](#)]
20. Mou, C.; Wang, Z.; Wells, D.R.; Xie, X.; Iliescu, T. Reduced order models for the quasi-geostrophic equations: A brief survey. *Fluids* **2021**, *6*, 16. [[CrossRef](#)]
21. Ravindran, S.S. Reduced-order adaptive controllers for fluid flows using POD. *J. Sci. Comput.* **2000**, *15*, 457–478. [[CrossRef](#)]
22. Pulgar-Painemal, H.A.; Sauer, P.W. Towards a wind farm reduced-order model. *Electr. Power Syst. Res.* **2011**, *81*, 1688–1695. [[CrossRef](#)]
23. Benner, P.; Schilders, W.; Grivet-Talocia, S.; Quarteroni, A.; Rozza, G.; Miguel Silveira, L. *Model Order Reduction: Volume 3 Applications*; De Gruyter: Berlin, Germany, 2020; p. 466.
24. Lall, S.; Marsden, J.E.; Glavaški, S. A subspace approach to balanced truncation for model reduction of nonlinear control systems. *Int. J. Robust Nonlinear Control IFAC-Affil. J.* **2002**, *12*, 519–535. [[CrossRef](#)]
25. Wang, Z.; Xiao, D.; Fang, F.; Govindan, R.; Pain, C.C.; Guo, Y. Model identification of reduced order fluid dynamics systems using deep learning. *Int. J. Numer. Methods Fluids* **2018**, *86*, 255–268. [[CrossRef](#)]
26. Fresca, S.; Manzoni, A.; Dedè, L.; Quarteroni, A. POD-enhanced deep learning-based reduced order models for the real-time simulation of cardiac electrophysiology in the left atrium. *Front. Physiol.* **2021**, *12*, 679076. [[CrossRef](#)] [[PubMed](#)]
27. Daniel, T.; Casenave, F.; Akkari, N.; Ryckelynck, D. Model order reduction assisted by deep neural networks (ROM-net). *Adv. Model. Simul. Eng. Sci.* **2020**, *7*, 1–27. [[CrossRef](#)]
28. Montanaro, A.; Pallister, S. Quantum algorithms and the finite element method. *Phys. Rev. A* **2016**, *93*, 032324. [[CrossRef](#)]
29. Chen, Z.Y.; Xue, C.; Chen, S.M.; Lu, B.H.; Wu, Y.C.; Ding, J.C.; Huang, S.-H.; Guo, G.P. Quantum Finite Volume Method for Computational Fluid Dynamics with Classical Input and Output. *arXiv* **2021**, arXiv:2102.03557.
30. Lloyd, S.; De Palma, G.; Gokler, C.; Kiani, B.; Liu, Z.-W.; Marvian, M.; Tennie, F.; Palmer, T. Quantum algorithm for nonlinear differential equations. *arXiv* **2020**, arXiv:2011.06571.
31. Liu, J.P.; Kolden, H.O.; Krovi, H.K.; Loureiro, N.F.; Trivisa, K.; Childs, A.M. Efficient quantum algorithm for dissipative nonlinear differential equations. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2026805118. [[CrossRef](#)] [[PubMed](#)]
32. Harrow, A.W.; Hassidim, A.; Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **2009**, *103*, 150502. [[CrossRef](#)]
33. Childs, A.M.; Kothari, R.; Somma, R.D. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM J. Comput.* **2017**, *46*, 1920–1950. [[CrossRef](#)]
34. Leyton, S.K.; Osborne, T.J. A quantum algorithm to solve nonlinear differential equations. *arXiv* **2008**, arXiv:0812.4423.
35. Gaitan, F. Finding flows of a Navier–Stokes fluid through quantum computing. *npj Quantum Inf.* **2020**, *6*, 1–6. [[CrossRef](#)]
36. Star, S.K.; Sanderse, B.; Stabile, G.; Rozza, G.; Degroote, J. Reduced order models for the incompressible Navier–Stokes equations on collocated grids using a ‘discretize-then-project’ approach. *Int. J. Numer. Methods Fluids* **2021**, *93*, 2694–2722. [[CrossRef](#)]
37. Sanderse, B. Non-linearly stable reduced-order models for incompressible flow with energy-conserving finite volume methods. *J. Comput. Phys.* **2020**, *421*, 109736. [[CrossRef](#)]
38. Molina-Aiz, F.D.; Fatnassi, H.; Boulard, T.; Roy, J.C.; Valera, D.L. Comparison of finite element and finite volume methods for simulation of natural ventilation in greenhouses. *Comput. Electron. Agric.* **2010**, *72*, 69–86. [[CrossRef](#)]
39. Fallah, N.A.; Bailey, C.; Cross, M.; Taylor, G.A. Comparison of finite element and finite volume methods application in geometrically nonlinear stress analysis. *Appl. Math. Model.* **2000**, *24*, 439–455. [[CrossRef](#)]
40. Turkel, E. Preconditioning techniques in computational fluid dynamics. *Annu. Rev. Fluid Mech.* **1999**, *31*, 385–416. [[CrossRef](#)]
41. Clader, B.D.; Jacobs, B.C.; Sprouse, C.R. Preconditioned quantum linear system algorithm. *Phys. Rev. Lett.* **2013**, *110*, 250504. [[CrossRef](#)]
42. Shao, C.; Xiang, H. Quantum circulant preconditioner for a linear system of equations. *Phys. Rev. A* **2018**, *98*, 062321. [[CrossRef](#)]
43. Giovannetti, V.; Lloyd, S.; Maccone, L. Quantum random access memory. *Phys. Rev. Lett.* **2008**, *100*, 160501. [[CrossRef](#)]
44. Ullmann, S.; Rotkvic, M.; Lang, J. POD-Galerkin reduced-order modeling with adaptive finite element snapshots. *J. Comput. Phys.* **2016**, *325*, 244–258. [[CrossRef](#)]
45. Luchtenburg, D.M.; Noack, B.R.; Schlegel, M. *An Introduction to the POD Galerkin Method for Fluid Flows with Analytical Examples and MATLAB Source Codes*; Technical Report 01/2009; Berlin Institute of Technology MB1: Berlin, Germany, 2009.
46. Lorenzi, S.; Cammi, A.; Luzzi, L.; Rozza, G. POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations. *Comput. Methods Appl. Mech. Eng.* **2016**, *311*, 151–179. [[CrossRef](#)]
47. Stabile, G.; Hijazi, S.; Mola, A.; Lorenzi, S.; Rozza, G. POD-Galerkin reduced order methods for CFD using Finite Volume Discretisation: Vortex shedding around a circular cylinder. *arXiv* **2017**, arXiv:1701.03424.
48. Rewienski, M.; White, J. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2003**, *22*, 155–170. [[CrossRef](#)]

49. White, J.K. A Trajectory Piecewise-Linear Approach to Model Order Reduction of Nonlinear Dynamical Systems. Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 2003.
50. Gratton, D.; Willcox, K. Reduced-order, trajectory piecewise-linear models for nonlinear computational fluid dynamics. In Proceedings of the 34th AIAA Fluid Dynamics Conference and Exhibit, Portland, OR, USA, 28 June–1 July 2004; p. 2329.
51. Striebel, M.; Rommes, J. Model order reduction of nonlinear systems: Status, open issues, and applications. In *Chemnitz Scientific Computing Preprints*; Springer: Berlin/Heidelberg, Germany, 2008.
52. Vasilyev, D.; Rewienski, M.; White, J. A TBR-based trajectory piecewise-linear algorithm for generating accurate low-order models for nonlinear analog circuits and MEMS. In Proceedings of the 2003 Design Automation Conference (IEEE Cat. No.03CH37451), Anaheim, CA, USA, 2–6 June 2003; pp. 490–495.
53. Voß, T. Model reduction for nonlinear differential-algebraic equations. Master’s Thesis, University of Wuppertal, Wuppertal, Germany, 2005.
54. Siegel, S.; Cohen, K.; Seidel, J.; McLaughlin, T. Short time proper orthogonal decomposition for state estimation of transient flow fields. In Proceedings of the 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 10–13 January 2005; p. 296.
55. Dao, M.H. Projection-Based Reduced Order Model for Simulations of Nonlinear Flows with Multiple Moving Objects. *arXiv* **2021**, arXiv:2106.02338.
56. Kornbleuth, M. Studying the Viscous Flow Around a Cylinder Using OpenFoam. Master’s Thesis, Boston University, Boston, MA, USA, 2016. [[CrossRef](#)]
57. Roshko, A. Experiments on the flow past a circular cylinder at very high Reynolds number. *J. Fluid Mech.* **1961**, *10*, 345–356. [[CrossRef](#)]
58. Karabelas, S.J.; Koumroglou, B.C.; Argyropoulos, C.D.; Markatos, N.C. High Reynolds number turbulent flow past a rotating cylinder. *Appl. Math. Model.* **2012**, *36*, 379–398. [[CrossRef](#)]
59. Perrin, R.; Braza, M.; Cid, E.; Cazin, S.; Barthet, A.; Sevrain, A.; Mockett, C.; Thiele, F. Obtaining phase averaged turbulence properties in the near wake of a circular cylinder at high Reynolds number using POD. *Exp. Fluids* **2007**, *43*, 341–355.
60. Borggaard, J.; Duggleby, A.; Hay, A.; Iliescu, T.; Wang, Z. Reduced-order modeling of turbulent flows. In Proceedings of MTNS, Blacksburg, VA, USA, 28 July–1 August 2008.
61. Balajewicz, M.J.; Dowell, E.H.; Noack, B.R. Low-dimensional modelling of high-Reynolds-number shear flows incorporating constraints from the Navier–Stokes equation. *J. Fluid Mech.* **2013**, *729*, 285–308. [[CrossRef](#)]
62. Kunisch, K.; Volkwein, S. Optimal snapshot location for computing POD basis functions. *ESAIM Math. Model. Numer. Anal.* **2010**, *44*, 509–529.
63. Lass, O.; Volkwein, S. Adaptive POD basis computation for parametrized nonlinear systems using optimal snapshot location. *Comput. Optim. Appl.* **2014**, *58*, 645–677. [[CrossRef](#)]
64. Li, K.; Huang, T.Z.; Li, L.; Lanteri, S. POD-based model order reduction with an adaptive snapshot selection for a discontinuous Galerkin approximation of the time-domain Maxwell’s equations. *J. Comput. Phys.* **2019**, *396*, 106–128.
65. Bond, B.; Daniel, L. Parameterized model order reduction of nonlinear dynamical systems. In Proceedings of the ICCAD-2005. IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, USA, 6–10 November 2005; pp. 487–494.
66. Lohmann, B.; Eid, R. Efficient order reduction of parametric and nonlinear models by superposition of locally reduced models. In *Methoden und Anwendungen der Regelungstechnik. Erlangen-Münchener Workshops 2007 und 2008*; Shaker Verlag: Aachen, Germany, 2009; pp. 27–36.
67. Zancanaro, M.; Mrosek, M.; Stabile, G.; Othmer, C.; Rozza, G. Hybrid neural network reduced order modelling for turbulent flows with geometric parameters. *Fluids* **2021**, *6*, 296.
68. Akkari, N.; Mercier, R.; Moureau, V. Geometrical reduced order modeling (ROM) by proper orthogonal decomposition (POD) for the incompressible Navier Stokes equations. In Proceedings of the 2018 AIAA Aerospace Sciences Meeting, Kissimmee, FL, USA, 8–12 January 2018; p. 1827.
69. Stabile, G.; Zancanaro, M.; Rozza, G. Efficient geometrical parametrization for finite-volume-based reduced order methods. *Int. J. Numer. Methods Eng.* **2020**, *121*, 2655–2682.