MOX-Report No. 53/2025

Telescopic quantum simulation of the advection-diffusion-reaction dynamics

Zecchi, A. A.; Sanavio, C.; Perotto, S.; Succi, S.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it https://mox.polimi.it

Telescopic quantum simulation of the advection-diffusion-reaction dynamics

Alessandro Andrea Zecchi[1], Claudio Sanavio[2], Simona Perotto[1] and Sauro Succi[2]

September 8, 2025

[1]Modellistica e Calcolo Scientifico (MOX),
Dipartimento di Matematica,
Politecnico di Milano,
Piazza L. da Vinci, 32, I-20133 Milano, Italy

[2]Center for Life Nano-Neuroscience at la Sapienza,
Fondazione Istituto Italiano di Tecnologia,
Viale Regina Elena 291, Roma, 00161, Italy

Abstract

Quantum singular value transformation (QSVT) is a powerful technique that applies a polynomial transformation to the singular values of a matrix encoded in a unitary operator. Many quantum algorithms can be viewed as a particular application of this technique, though its application to the quantum computation of classical dynamics has not been extensively explored. We introduce a telescopic quantum algorithm for solving the advection-diffusion-reaction (ADR) equation at a given (possibly large) final simulation time, applying the QSVT to an efficient block-encoding of the matrix representing the considered dynamics. We decompose the ADR time evolution function using a Chebyshev polynomial of degree d and we show that effectively exploiting the spectral knowledge of the input matrix within the QSVT protocol can provide a similar simulation error with up to an order of magnitude lower of polynomial degree. The associated quantum circuit employs only $n + 4$ qubits where $N = 2^n$ is the number of spatial discretization points, and achieves circuit depth of $\mathcal{O}(d \times \text{poly}(n))$. The efficient use of quantum resources and the reduced overall complexity pave the way for the application of the proposed algorithm on near term quantum hardware.

1 Introduction

The numerical simulation of transport phenomena in fluid dynamics is one of the major challenges for many practical applications in science and engineering due to the associated computational cost. Recent advances in quantum computing provide novel approaches that offer a significant speed-up when compared to classical methods [1, 2, 3]. Developing quantum algorithms for computational fluid dynamics is an active field

of research. Among the most promising strategies are a quantum version of the lattice gas cellular automata [4, 5, 6] and of the lattice Boltzmann method [7, 8, 9].

The potential for such advantage derives from peculiar features of quantum mechanics, such as superposition and entanglement, which offer powerful tools that, when harnessed by carefully designed algorithms, can surpass conventional computational methods. These principles allow quantum computers to perform linear algebra computations in a state space that grows exponentially with the number of its information units, the qubits [10]. The quantum singular value transformation (QSVT) provides a powerful framework for applying polynomial transformations to the singular values of a matrix encoded within a unitary operator [11]. This technique has already been used for different applications, from matrix inversion [12, 13] to Hamiltonian simulation [14] among others [15, 16]. In this article we apply QSVT to the simulation of classical transport phenomena with the aim of developing new techniques for the practical realization of quantum computational fluid dynamics algorithms, effectively reducing the quantum resource requirements for a complete simulation. In particular, in section 2 we review the main mathematical and practical aspects of QSVT. Subsequently, in Section 3 we introduce a new algorithm based on QSVT to simulate the unsteady advection-diffusion-reaction (ADR) equation for a one-dimensional test case, we analyse the associated error, the circuit complexity and the success probability of the algorithm. Using numerical simulations we validate the proposed algorithm in Section 4. Finally, we comment on the key findings and discuss future developments in Section 5.

2 QSVT

By the late 20th century, quantum computers were proposed as a revolutionary tool for simulating quantum physics [17]. Subsequently, the physical realization of quantum computing hardware has rapidly progressed, and numerous quantum algorithms have been developed able to offer a significant speedup over their classical counterparts. Notable examples include Shor's algorithm for integer factorization, Grover's algorithm for searching unstructured databases and the HHL algorithm for solving linear systems of equations. QSVT, recently developed in a pioneering work [11], is a generalization to higher dimensions of quantum signal processing (QSP) and qubitization [18, 19], and offers a unifying perspective of many different algorithms. It originates from the study of the product of two reflections which can be interpreted as a rotation in the two dimensional space where the overall operation can be decomposed. This feature is shared by many successful quantum computing algorithms, such as the Grover's search algorithm, and the Szegedy's quantum walk algorithm [20, 21].

In this work, we use the QSVT to apply a polynomial transformation to the singular values of a projected matrix A embedded in a unitary matrix U_A. A simple way to construct projected unitary encodings is to use the block-encoding technique [19, 22]. We report the main results from Refs. [11] and [15] by introducing the following definition for the singular value decomposition (SVD) of a projected unitary matrix.

Definition 2.1 (SVD of a projected unitary matrix) *Let $A \in \mathbb{C}^{m\times n}$ be a generic matrix, then there exist two unitary matrices $W \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$ such that*

$$A = W\Sigma V^{\dagger} \tag{1}$$

where $\Sigma = diag(\sigma_1, \cdots, \sigma_p) \in \mathbb{R}^{m\times n}$ with $p = min(m, n)$ and $\sigma_1 \geq \cdots \geq \sigma_p \geq 0$ are the singular values of A.

If A is embedded in a unitary operator U_A, then there exist the orthogonal projectors Π and $\tilde{\Pi}$ such that $A = \tilde{\Pi}U_A\Pi$. Moreover, there exist two orthonormal bases $|v_i\rangle$ and $|w_i\rangle$ of the subspaces $Im(\Pi)$ and $Im(\tilde{\Pi})$ respectively, such that

$$A = \sum_{i=1}^{p} \sigma_i \left|w_i\right\rangle \left\langle v_i\right| . \tag{2}$$

Using an alternating phase modulation circuit $U_{\vec{\Phi}}$ defined, for a given set of d phase factors $\vec{\Phi} = [\phi_1, \phi_2, \ldots, \phi_d]^T \in \mathbb{R}^d$, as

$$U_{\vec{\Phi}} = \begin{cases} U_A e^{i\phi_d(2\Pi - I)} ... U_A^{\dagger} e^{i\phi_2(2\tilde{\Pi} - I)} U_A e^{i\phi_1(2\Pi - I)} & \text{for odd } d \\ U_A^{\dagger} e^{i\phi_d(2\tilde{\Pi} - I)} ... U_A^{\dagger} e^{i\phi_2(2\tilde{\Pi} - I)} U_A e^{i\phi_1(2\Pi - I)} & \text{for even } d, \end{cases} \tag{3}$$

then, for a generic matrix A embedded in a unitary operator U_A, the following theorem on singular value transformation by a polynomial function, denoted generically as $P^{SV}(A)$, holds [11]:

Theorem 2.1 (QSVT by alternating phase modulation) *Let $\vec{\Phi} \in \mathbb{R}^d$ be a set of d phase factors, let $P : \mathbb{R} \to \mathbb{C}$ be a polynomial of degree at most d and let U_A be the projected unitary matrix encoding of a matrix A, then for odd d, the polynomial P is odd and*

$$P^{SV}(A) = \sum_{i=1}^{p} P(\sigma_i) \left|w_i\right\rangle \left\langle v_i\right| = \tilde{\Pi}U_{\vec{\Phi}}\Pi \tag{4}$$

which consists in a polynomial transformation to the singular value of A. Similarly, for an even value of d, the polynomial P is even and it holds

$$P^{SV}(A) = \sum_{i=1}^{p} P(\sigma_i) \left|v_i\right\rangle \left\langle v_i\right| = \Pi U_{\vec{\Phi}}\Pi \tag{5}$$

which consists in a polynomial transformation to the singular value of A with a modification of the input and output spaces.

Using this result and the following Corollary 11 of [11] it can be showed that if the polynomial P is real valued, has parity d mod 2 and $|P(x)| \leq 1 \ \ \forall x \in [-1, 1]$, then:

$$P^{SV}(A) = \begin{cases} (\langle +| \otimes \tilde{\Pi})(|0\rangle \langle 0| U_{\vec{\Phi}} + |1\rangle \langle 1| U_{-\vec{\Phi}})(|+\rangle \otimes \Pi) & \text{for odd } d \\ (\langle +| \otimes \Pi)(|0\rangle \langle 0| U_{\vec{\Phi}} + |1\rangle \langle 1| U_{-\vec{\Phi}})(|+\rangle \otimes \Pi) & \text{for even } d. \end{cases} \tag{6}$$

It is worth noting that it is possible to reformulate the previous results using appropriate QSP convention for the phase factors [15]. In this case, the phase factors $\vec{\Phi}$ are $d+1$ real parameters and the alternating phase modulation circuit can be defined accordingly to QSP definitions. Moreover, the quantum circuit associated with Eq. (6), can be reformulated with a reduced complexity: the two controlled versions of the alternating phase modulation circuit (with opposite angles) are equivalent to using an Hadamard gate on the ancillary qubit employed for QSVT, while measuring the $|+\rangle$ state corresponds to measuring the $|0\rangle$ state with a final additional Hadamard gate. The improved circuit, using a minimum number of ancillary qubits, is presented in Figure 1, while the simplest implementation, i.e., when $\Pi = |0\rangle\langle 0|$, of a projector-controlled phase shift $\mathrm{PS}(\phi)$ which applies $e^{i\phi(2\Pi - I)}$ for a given angle ϕ, is shown in Figure 2.

Many applications of QSVT involve applying a real function $f : \mathbb{R} \to \mathbb{R}$ to a matrix. Some examples include the matrix inversion problem [12, 23], eigenvalue threshold filtering [15] and finding a Gibbs distribution for a given Hamiltonian [16]. In these cases a proper even or odd, limited polynomial approximation $p_d(x) \simeq f(x)$ of degree d of the target function should be implemented, and the corresponding phase factors $\vec{\Phi}$ have to be precisely computed. In particular, finding the phase factors to achieve the desired polynomial transformation is not an easy task, and different numerical methods can be used. An efficient iterative numerical method, using QSP symmetric conventions, described in [24] and further analyzed in [25], employs an expansion in terms of Chebyshev polynomials of the first kind of a target function and a cost function $L(\vec{\Phi})$ measuring the mean-squared loss between the target expansion and the alternating phase modulation circuit over discrete sample points. In order to minimize such a cost function, an L-BFGS quasi-Newton method is used to find the phase factors, exploiting symmetry, until a specific tolerance ϵ is reached, namely $L(\vec{\Phi}) \leq \epsilon$. We will refer to this method as symmetric-QSP. This optimization-based method is numerically stable, and it can be proved that the cost function has a global minimum $L(\vec{\Phi}^*) = 0$ (see Theorem 1 in [24]).

In addition, it is also important to remark that when A is a square $n \times n$ Hermitian matrix, the overall description is extensively simplified and the QSVT technique can be considered as a quantum eigenvalue transformation (QET) of the input matrix

$$P^{SV}(A) = \sum_i P(\lambda_i) |v_i\rangle \langle v_i| . \tag{7}$$

In fact, in this case all eigenvalues λ_i are real and the singular values are such that $\sigma_i = |\lambda_i|$. Moreover, the singular value decomposition is connected to the eigenvalue decomposition through the following relation [26]

$$A = \sum_i \lambda_i |v_i\rangle \langle v_i| = \sum_i |\lambda_i| \, |\mathrm{sign}(\lambda_i) v_i\rangle \langle v_i| = \sum_i \sigma_i |w_i\rangle \langle v_i| . \tag{8}$$

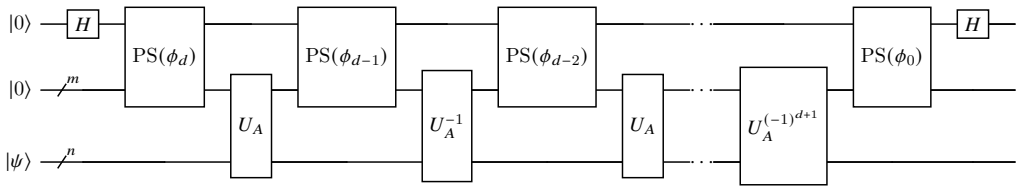


Figure 1: Full QSVT circuit employing the block-encoding U_A and its inverse $U_A^{-1} = U_A^\dagger$, to implement a real polynomial function of matrix A, when using $d+1$ phase factors. The circuit uses only 1 additional ancilla qubit beyond the register needed for the unitary block encoding U_A.

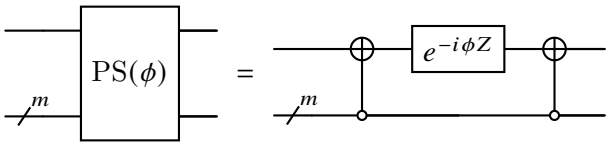


Figure 2: Projector-controlled phase shift PS(ϕ) implementation with a single ancilla qubit, using the control state $|0\rangle$, as is standard for block encoding.

3 QSVT applied to the advection-diffusion-reaction equation

We introduce here the advection-diffusion-reaction (ADR) equation, which is a prototypical model for one-dimensional transport, given by

$$\begin{cases} \dfrac{\partial \psi}{\partial t} = D\dfrac{\partial^2 \psi}{\partial x^2} - c\dfrac{\partial \psi}{\partial x} - a\psi & x \in (0, L), t > 0 \\ \psi(x, 0) = \psi_0 & x \in (0, L) \\ \psi(0, t) = \psi(L, t) & t > 0 \end{cases} \tag{9}$$

where ψ is the physical quantity of interest, whereas parameters $D, a \in \mathbb{R}^+$ and $c \in \mathbb{R}$ representing the constant diffusion, reaction and velocity coefficients, respectively. The problem has periodic boundary conditions and an initial condition ψ_0 is given.

Using a second order centered finite difference scheme to approximate the derivatives with respect to the variable x on a uniform distribution of spatial nodes $x_1 = 0 < x_2 < ... < x_N < x_{N+1} = L$, problem (9) can be discretized as:

$$\dot{\psi_j} = \frac{D}{\Delta x^2}(\psi_{j-1} - 2\psi_j + \psi_{j+1}) - \frac{c}{2\Delta x}(\psi_{j+1} - \psi_{j-1}) - a\psi_j \quad \text{for } j = 1, ..., N \tag{10}$$

with $\Delta x = L/N$ the uniform grid spacing and where $\dot{\psi}_j = \dfrac{d\psi_j}{dt} = \dfrac{d\psi(x_j, t)}{dt}$. The corresponding algebraic form is given by the following system of linear, first-order ordinary

differential equations

$$\begin{cases} \dot{\vec{\psi}}(t) = A\vec{\psi}(t) \\ \vec{\psi}(0) = \vec{\psi}_0 \end{cases} \tag{11}$$

with $A \in \mathbb{R}^{N\times N}$ and $\vec{\psi}(t) \in \mathbb{R}^N$. Without loss of generality, we take $N = 2^n$ where n is the number of qubits used to embed the numerical problem into a quantum register. Matrix A can be written in a compact form by highlighting the corresponding tridiagonal circulant structure being

$$\begin{aligned} A_{ij} = &\left(-\frac{2D}{\Delta x^2} - a\right)\delta_{i,j} + \left(\frac{D}{\Delta x^2} - \frac{c}{2\Delta x}\right)\delta_{i,j+1} \\ &+ \left(\frac{D}{\Delta x^2} + \frac{c}{2\Delta x}\right)\delta_{i,j-1} \quad \text{for } i,j = 1,\dots N. \end{aligned} \tag{12}$$

Matrix A can be efficiently block encoded with only 3 ancilla qubits, using the sparse access method described in [22] and subsequently employed in [27, 28]. The overall circuit that block encodes $A/4$ is provided in Figure 3.

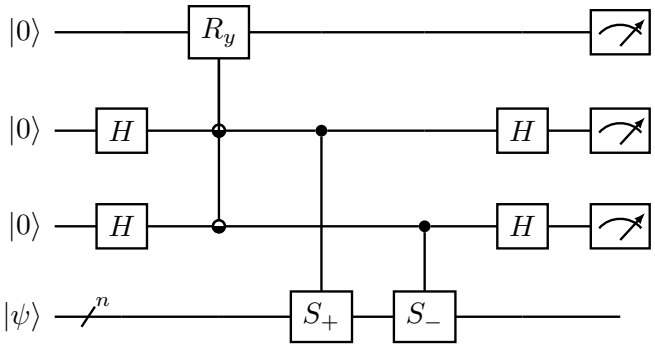


Figure 3: Quantum circuit for the implementation of matrix A in (12) using block-encoding. The initial state $|\psi\rangle$ is embedded in the n-qubit register, while the 3 ancilla qubits $|0\rangle^{\otimes 3}$ are used to embed the position and value of the non-null elements of the matrix A. The implementation uses a multi-controlled rotation R_y with a different angle for each value of the control qubits and two controlled shift operators S_+, S_- that increase and decrease by one the value of the target qubits respectively.

The exact solution of the initial value problem (11) is $\vec{\psi}(t) = e^{At}\vec{\psi}_0$. When the order of matrix A becomes larger, the cost of calculating this exponential can quickly become prohibitive on classical computers when time-discretization schemes are employed.

We adopt the following simplification for the problem under study: we use a reference frame moving at velocity c, which is equivalent to set $c = 0$. In this constant velocity case, the exact final state at a time t can be recovered by shifting the resulting state by a value $c \cdot t$ along the x interval. With this choice, the matrix A is symmetric and negative definite and the corresponding set of eigenvalues can be computed analytically

as

$$\lambda_i = \left(-\frac{2D}{\Delta x^2} - a\right) + \frac{2D}{\Delta x^2}\cos\left(\frac{2\pi i}{N}\right) \quad i = 0, 1, 2, \ldots, N-1. \tag{13}$$

More generally, the interval J where all the eigenvalues of A lie can be computed using Gershgorin circle theorem, being

$$J = \left[-\frac{4D}{\Delta x^2} - a, -a\right]. \tag{14}$$

In this case, the following quantum algorithm can be used to solve problem (11): given an efficient block-encoding of matrix A in a unitary matrix U_A [27], and a final time t, construct a QSVT circuit to compute a polynomial approximation of $f(x) = e^{-t|x|}$, where x is an eigenvalue of A. We remark that the efficient quantum circuit for matrix A is shown in Figure 3 and it represent the diffusion-reaction dynamics obtained by setting the velocity $c = 0$. The resulting operator is then applied to the initial state $\vec{\psi}_0$, embedded in the state of a target register using amplitude encoding

$$\vec{\psi}_0 \rightarrow |\psi_0\rangle = \frac{\sum_{j=1}^{N} \psi_{0,j} |j\rangle}{\|\vec{\psi}_0\|}.$$

The final state provides an approximation of $\vec{\psi}(t)$, up to a normalization factor, and is obtained conditionally on measuring the $|0\rangle$ state on the ancillary qubits employed. The advection part can be recovered by a simple shift of $c \cdot t$ using classical postprocessing or via an appropriate quantum circuit.

As a consequence of the previous analysis, we have chosen the function

$$f(x) = e^{-t|x|} \tag{15}$$

defined in the domain $[-1, 1]$, as the target function for a given time t, as is equivalent to the original exponential function for the negative eigenvalues of the input matrix A under the quantum eigenvalue transformation framework (see Equation (7)). Function f in (15) is the same used in [15] to approximate the Gibbs distribution $\rho(\beta) = 1/Ze^{-\beta\mathcal{H}}$ for a given Hamiltonian $\mathcal{H}$, an inverse temperature β and with partition function $Z = Z(\beta)$ by exploiting its definite parity. With this choice $f(x)$ is an even limited function, therefore there exists an approximating polynomial of degree d $p_d(x) \simeq f(x)$ satisfying the same requirements stated for Equation (6). We remark that this simplification can yield a quantum algorithm with a reduced circuit depth and fewer qubits than techniques based on time discretization, which obtain solutions at multiple time steps through matrix inversion [3].

It is worth mentioning that matrix A has to be properly scaled in order to be block-encoded, or alternatively the ADR coefficients c, D, a have to be scaled, so that matrix A has all the eigenvalues with modulus less than or equal to 1. Moreover, the block-encoding scheme may introduce an additional scaling factor. If the operator is represented as $A/\alpha = \tilde{\Pi} U_A \Pi$, then the simulation time t has to be rescaled as $t \rightarrow \alpha t$ so that the effect of alpha cancels out and $e^{-|\frac{A}{\alpha}|\alpha t} = e^{-|A|t}$.

We now turn to the problem of constructing an approximating polynomial for the function $f(x)$. It is well known that every continuous function on a bounded interval can be approximated to arbitrary accuracy by polynomials (Weierstrass approximation theorem [29]). A possible choice for the polynomial approximation $p_d \in \mathbb{P}_d$, of degree at most d, of $f(x)$ is obtained by using an expansion in terms of Chebyshev polynomials of the first kind, being

$$p_d(x) = \sum_{j=0}^{d} c_j T_j(x) \simeq f(x) \tag{16}$$

with $d \in \mathbb{N}$ the maximum degree of the expansion, $c_j \in \mathbb{R}$ the weight associated with the j-th polynomial $T_j(x) = \cos[j \arccos(x)]$. This choice has the additional advantage of being compatible with the efficient numerical method for computing the phase factors previously described [24]. The coefficients c_j are determined in order to find the best approximating polynomial which minimizes the error in the interval $[-1, 1]$, which is the domain of the polynomial transformation applied by QSVT to the input singular values. To this aim it is possible to use the Chebyshev discrete transform method, or a least squares fitting of Chebyshev series to the given data, or the Remez method [30, 24, 31].

3.1 Error analysis and improvement

The result of the algorithm, for a given time t, is the normalized state

$$|\psi(t)\rangle = C \cdot p_d(A) |\psi_0\rangle , \tag{17}$$

while the exact normalized solution is given by

$$|\psi_{exact}(t)\rangle = C' \cdot e^{At} |\psi_0\rangle . \tag{18}$$

Assuming that the norm factor is equal for both states, i.e., $C = C' = \| e^{At} |\psi_0\rangle \|_2^{-1}$, a sound assumption if the polynomial approximation is accurate, the Euclidean norm of the error can be written as

$$\| |\psi(t)\rangle - |\psi_{exact}(t)\rangle \|_2 = C \cdot \| (p_d(A) - e^{At}) |\psi_0\rangle \|_2 \tag{19}$$

where $\| \cdot \|_2$ denotes the Euclidean vector norm and the induced matrix spectral norm. Since matrix A is symmetric, it follows that also $p_d(A)$ and e^{At} are symmetric matrices. By exploiting the compatibility between the Euclidean and spectral norms, it follows that

$$\| |\psi(t)\rangle - |\psi_{exact}(t)\rangle \|_2 \leq C \| p_d(A) - e^{At} \|_2 \| |\psi_0\rangle \|_2 = C \cdot \rho(p_d(A) - e^{At}), \tag{20}$$

the initial quantum state being normalized (i.e., $\| |\psi_0\rangle \|_2 = 1$), and where $\rho(p_d(A) - e^{At})$ denotes the spectral radius of the difference between the two matrices $p_d(A)$ and e^{At}. Employing the QET framework (see Equation (7)) we have that

$$\rho(p_d(A) - e^{At}) = \max_{i=1,\ldots,N} |p_d(\lambda_i) - e^{\lambda_i t}| \leq \sup_{x \in J} |p_d(x) - f(x)| = \| p_d - f \|_{\infty, J}, \tag{21}$$

where J is the interval containing all the eigenvalues of A as defined in Equation (14), and where, from now on, $\|\cdot\|_{\infty,J}$ denotes the supremum norm restricted to the interval J. The time dependence of functions f and p_d is not made explicit. Thus, we can conclude that

$$\| \, |\psi(t)\rangle - |\psi_{exact}(t)\rangle \, \|_2 \leq C \|p_d - f\|_{\infty,J}. \tag{22}$$

Therefore, disregarding any inaccuracy introduced by the calculation of the phase factor, the main computational source of error in the proposed algorithm to solve (11) is given by the approximation error $\|p_d - f\|_{\infty,J}$, which depends on the degree d of the polynomial expansion and on the final time t.

There are different possible choices to find the best Chebyshev polynomial approximation of degree d of function f. A straightforward approach is to use the least-squares fitting of function f at the M Chebyshev nodes

$$x_i = -\cos(\pi i/M), \quad i = 0, ..., M-1 \tag{23}$$

over $[-1, 1]$, with $M \geq d+1$. Thus, using a Chebyshev expansion of degree d

$$p_d(x) = c_0 + c_1 T_1(x) + c_2 T_2(x) + \cdots + c_d T_d(x) \simeq f(x), \tag{24}$$

the $d+1$ unknowns c_i collected in vector $\vec{c}$, which minimize $\sum_i |p(x_i) - f(x_i)|^2$, can be found by solving the system (possibly over-determined) of equations

$$V\vec{c} = \vec{y} \tag{25}$$

where $\vec{y} = [f(x_0), f(x_1), ..., f(x_{M-1\}})]^T$ is the function evaluation vector and $V \in \mathbb{R}^{M\times(d+1)}$ is the pseudo Vandermonde matrix associated with the Chebyshev polynomials.

As an alternative to the least-squares fitting, in [24], the authors use the Remez method to build polynomial p_d, that converges uniformly to the best approximating polynomial p_d^* with linear or quadratic convergence, depending on the regularity of function $f(x)$.

We propose a improvement to this framework by using an approximating polynomial of degree d designed to achieve higher accuracy not over the entire interval $[-1, 1]$ but specifically on the relevant subset $[-1, -a]$ which contains the interval J where the eigenvalues of matrix A lie. This choice can lead to a reduced error using the same degree d (see Figure 4) or similar error by using a lower degree approximating polynomial. An even smaller error could be achieved by by restricting the approximation to the interval J. However, we propose to extend the interval to -1 on the left since this prevents the resulting polynomial from exhibiting rapid oscillations as typically occurs in Runge's phenomenon [30]. As a result, the polynomial can be reliably used to compute the phase factors through the symmetric-QSP method [24]. In particular, we want to solve the following constrained optimization problem: for a given time t and an assigned degree d, find the best approximating polynomial, $\hat{p}_d^*$ expressed as an expansion in even Chebyshev polynomials of the first kind, on the interval $[-1, -a]$ such that

$$\begin{cases} \hat{p}_d^* = \underset{p_d \in \mathbb{P}_d}{\operatorname{argmin}} \|p_d - f\|_{\infty,J} \\ |\hat{p}_d^*(x)| \leq 1 \quad \forall x \in [-1, 1]. \end{cases} \tag{26}$$

We remark that the Remez method cannot be used to solve this problem as it may provide an approximating polynomial which does not satisfy the constraint $|\hat{p}_d(x)| \leq 1 \quad \forall x \in [-1, 1]$. Problem (26) can be numerically solved using efficient libraries [32, 33] that offer high level interfaces for formulating and solving convex optimization problems in order to obtain an approximate solution that we denote by $\hat{p}_d$. This polynomial can provide a significantly lower absolute error (i.e., a simulation error) compared to a polynomial p_d which has been computed to approximate function f on the full interval $[-1, 1]$, namely it holds

$$\|\hat{p}_d - f\|_{\infty,J} \leq \|p_d - f\|_{\infty,J}, \tag{27}$$

for any polynomial p_d of degree d defined as in (16), approximating f on the whole interval $[-1, 1]$. In Section 4, we will show that the $\hat{p}_d$ case achieves experimentally comparable errors to p_d, while requiring up to an order of magnitude lower polynomial degree and, consequently, fewer phase factors.

3.2 Circuit complexity and success probability

Ad discussed before, the sparse banded-circulant matrix in Equation (12) can be efficiently block-encoded into an operator U_A using $n+3$ qubits and $O(\text{poly}(n))$ gates, while the QSVT implementation of a real polynomial transformation requires one additional ancillary qubit. Using a degree d polynomial expansion leads to an overall circuit equivalent to the one shown in Figure 1. This circuit consists of a repeated application of operator U_A and its inverse, interleaved with $d+1$ controlled phase factors $e^{i\phi_i(2\Pi - I)}$, acting as projectors. Hence, the total circuit complexity scales as $O(d \times \text{poly}(n))$ and is therefore considered efficient. The quantum circuit can be viewed as a block encoding of e^{At} applied to the initial state $|\psi_0\rangle$. As a consequence, the success probability of the algorithm is given by

$$p(0) = c_0^2 \| p_d(A) |\psi_0\rangle \|_2^2 \simeq c_0^2 \| e^{At} |\psi_0\rangle \|_2^2, \tag{28}$$

where c_0 is a constant that depends on A and t, while $p_d(A)$ is the $d-$degree polynomial transformation of the input matrix. Notice that the desired simulation result is obtained conditionally on measuring the ancillary qubits in the $|0\rangle$ state. When the reaction term in (12) is dominant, the success probability rapidly decays as e^{-2at}, while in a scenario characterized by a dominant diffusion the success probability stays close to c_0^2 as in this case the term e^{At} is close to a unitary operator (norm preserving). From numerical simulations it is possible to find that $c_0^2 \simeq 1$ as discussed in next sections.

3.3 Proposed algorithm

In conclusion, the overall algorithm (described in Algorithm 1) is significantly improved by using a polynomial approximation of the function f in (15), restricted to the subinterval of interest $[-1, -a]$. The algorithm outputs a quantum state corresponding to an approximate solution of the discretized ADR problem (11) at a given final time t. The circuit size of the proposed algorithm scales efficiently with the number of qubits but

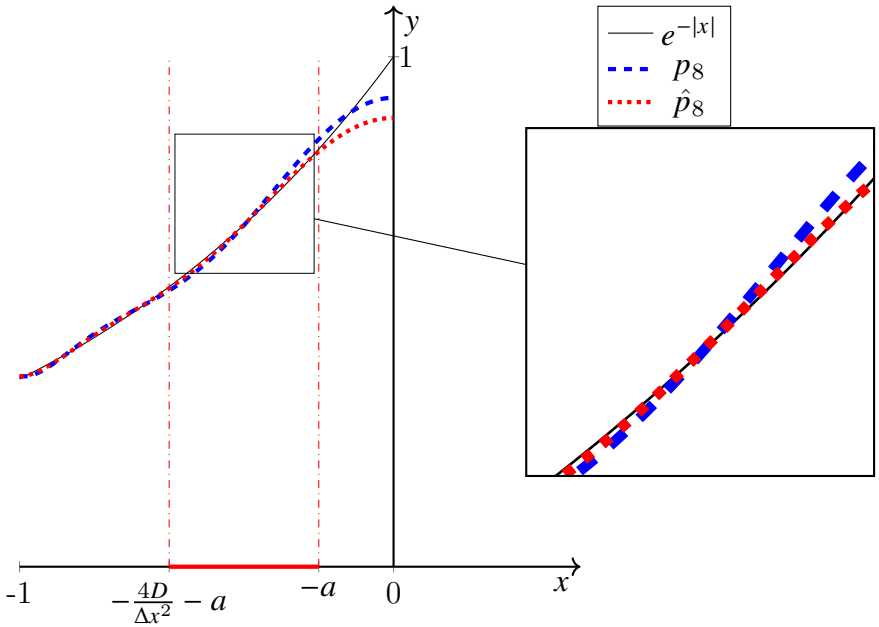


Figure 4: The eigenvalues of a a proper scaling of matrix A in (12), with $c = 0$ and D and a positive constants, lie inside the closed interval $J = [-\frac{4D}{\Delta x^2} - a, -a]$ centered at $-\frac{2D}{\Delta x^2} - a$. All the represented functions are symmetric in $[-1, 1]$, but only the left part of the interval is shown. The black curve represents the transformation $e^{-|x|}$ that we aim to apply to the input eigenvalues for $t = 1$. The dashed curve denotes a possible polynomial approximation of degree $d = 8$ of the target function, computed over the entire interval $[-1, 1]$, while the dotted curve $\hat{p}_8$ is a polynomial approximation of the same function again of degree $d = 8$, but targeted to minimize the error in the interval $[-1, -a]$ and with an additional constraint of a limited range. In the enlarged view on the right it is possible to appreciate the better approximation of the desired curve provided by $\hat{p}_8$ in J when compared with p_8.

has the drawback of requiring a higher polynomial degree d (and thus a deeper circuit) in order to simulate longer times while maintaining the same level of accuracy.

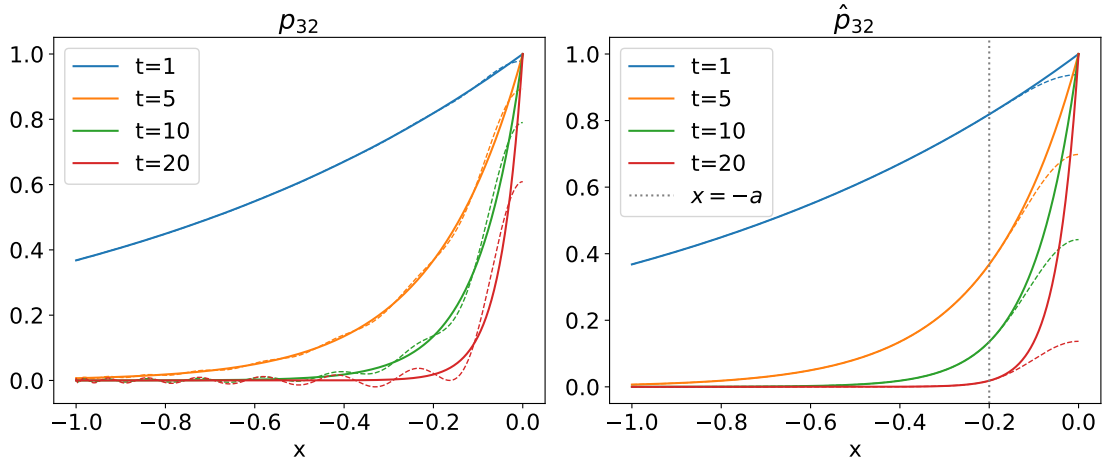
The characteristic times for diffusion $\tau_{diff} = L^2/D$, reaction $\tau_{re} = 1/a$, and advection $\tau_{adv} = L/c$ define the relevant time scales. For a meaningful simulation, the total simulated time should extend over an interval comparable to these values to capture the essential dynamics. If we consider diffusion, the time scale can be expressed as $\tau_{diff} = (\Delta x N)^2/D$, which scales as $O(N^2)$ where $N = 2^n$. This scaling poses a challenge for the proposed algorithm since simulating longer times requires a higher polynomial degree d to achieve the desired transformation within a given error ϵ. In fact computing a polynomial approximation of the function $f(x) = e^{-t|x|}$ is a challenging task when the time t increases. However, restricting the simulation time to values on the order of a few cell diffusion times $\tau_d = \Delta x^2/D$, which in our case so that $|-4D/\Delta x^2 - a| \leq 1$, can be achieved with a polynomial degree of about $d \simeq 100$ while maintaining a reasonable error. An intermediate goal should is to reach a simulation time scale of $O(N)$. Achieving such goal would provide an exponential advantage over classical computation for the simulation of classical systems with geometrically local interactions such as advection-diffusion-reaction dynamics [34].

Input: matrix A with velocity parameter set to $c = 0$, time t, initial state $|\psi_0\rangle$, even degree d, velocity c
Output: Final state $|\psi(t)\rangle$
if $\|A\|_2 > 1$ **then**
  $\alpha = \|A\|_2$;
  $A = A/\alpha$;
end
$U_A \leftarrow$ block encoding of A; $\beta \leftarrow$ scaling factor of U_A;
$\alpha = \alpha \times \beta$;
$t = \alpha \times t$;
$[c_0, c_1, c_2, \ldots, c_d]^T \leftarrow$ solve problem 26 for $\hat{p}_d$;
$\vec{\Phi} = [\phi_0, \phi_1, \ldots, \phi_d]^T \leftarrow$ use quasi-Newton symmetric-QSP method for the coefficients $[c_0, c_1, c_2, \ldots, c_d]^T$;
$r = 1$;
while $r \neq 0$ **do**
  $|\cdot\rangle\,|\psi(t)\rangle = QSVT(U_A, U_A^\dagger, \vec{\Phi})\,|0\rangle\,|\psi_0\rangle$;
  $m \leftarrow$ measure the state of 4 ancillary qubits;
  if $m = |0\rangle$ **then**
    $r = 0$;
    Shift the resulting state by the factor $c \cdot t$ to recover the advection part.
  end
end

Algorithm 1: QSVT for advection-diffusion-reaction problems with constant coefficients

4 Numerical results

In this section we present the results of numerical simulations carried out using the Qiskit software framework [35] and the `pyqsp` library [24] for computing the phase factors , which implements the quasi-Newton symmetric-QSP method. The `cvxpy` library was also employed to solve polynomial approximation problems. The results have also been validated using spectral transformations of the input matrix A.



(a) (b)

Figure 5: Comparison between polynomials of degree $d = 32$ (dashed lines) approximating function $f(x) = e^{-|x|t}$ (solid lines) (a) in the interval $[-1, 1]$ and (b) in the interval $J = [-1, -a]$ for different times t. The vertical dotted line in (a) represents the right endpoint of the J interval.

In Figure 5, we compare the polynomial approximation p_d of function f using standard least squares fitting over $[-1, 1]$ with the improved polynomial approximation $\hat{p}_d$ restricted to the interval $[-1, -0.2]$ obtained solving problem (26) for a degree $d = 32$. The improved polynomial $\hat{p}_d$ accurately approximates the target function over the required interval with a significantly reduced error.

The quantum circuit for QSVT has been implemented using Qiskit, and is the one represented in Figure 1 where the number of grid sites is $N = 2^n$ with n working qubits, while the block encoding of A employs 3 ancillary qubits. The circuit depth scales as $O(d \times \text{poly}(n))$ with $\text{poly}(n) \simeq n^{3.7}$ as verified by transpiling the circuit for a generic quantum hardware (see Figure 6). This scaling can be considered efficient and primarily depends on the block encoding method employed for the matrix A (see Figure 3). The success probability is close to 1 for the pure diffusive case, indicating that $c_0^2 \simeq 1$ in Equation (28). To demonstrate the effectiveness of the proposed method, we have measured the Euclidean error between the normalized solution obtained from the quantum simulation and the exact solution, considering the two different cases of polynomial

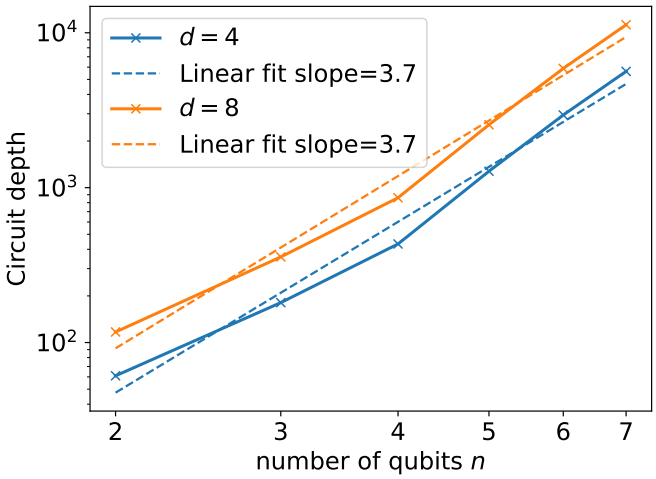


Figure 6: Circuit depth (solid lines) as a function of the number of working qubits n with degree $d = 4$ and $d = 8$ of polynomial expansion for QSVT, using a logarithmic scale for both axes. The interpolating dashed lines correspond to a polynomial scaling of $n^{3.7}$.

approximation, p_d and $\hat{p}_d$, of the function $f(x) = e^{-t|x|}$. For the simulation we use $n = 6$ working qubits while we set $\frac{D}{\Delta x^2} = 0.2$ and $a = 0.1$. The initial state of the system is the Gaussian function as depicted in Figure 8 for $t = 0$. For the sake of simplicity, we have used a block encoding with scaling factor 1 which does not require rescaling the simulation time. In Figure 7 the solid lines show the error while the dashed lines represent the error bound of Equation (22) for the simulation times $t = 2$ and $t = 10$ which are measured in units of cell diffusion time $\tau_d = \Delta x^2/D$. These results indicate that employing a polynomial approximation of f on the interval $[-1, -a]$ can reduce the required degree to achieve a target error tolerance by up to an order of magnitude. For a qualitative comparison, Figure 8 shows the simulation results, adjusted for the advection component, against the exact solution at different times using polynomial expansion of degree $d = 100$, starting from an initial Gaussian profile. Figure 8 also shows that, for a fixed polynomial degree, the accuracy of the simulation decreases as times increases. Finally, a comprehensive representation of the error as a function of time and the degree of polynomial approximation $\hat{p}_d$ over the interval $[-1, -a]$, is provided in Figure 9, for $N = 64$ sites, $\frac{D}{\Delta x^2} = 0.2$, and $a = 0.1$.

5 Conclusions and discussion

The ADR equation serves as a fundamental model for the analysis of transport phenomena in classical fluid systems. When the ADR coefficients are constant and the model setting is one-dimensional, the dynamics can be solved analytically. However, when the model is multidimensional and characterized by complex boundary conditions, the solu-

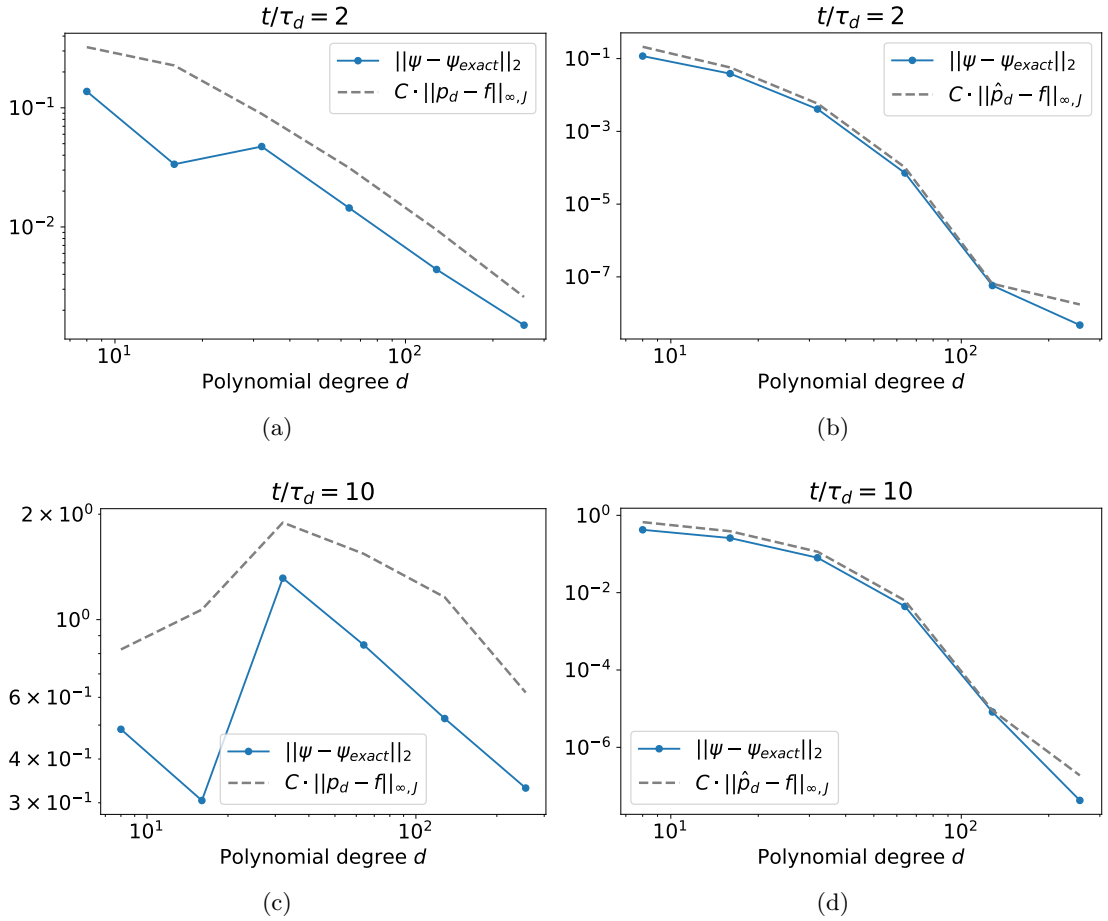


Figure 7: Simulation error (solid lines) in the Euclidean norm between the exact solution and the QSVT result shown on logarithmic axes. Panels on the left (b) $t = 2\tau_d$ and (d) $t = 10\tau_d$ for the standard case where the polynomial p_d has been computed to approximate function f on the full interval $[-1, 1]$. Panels on the right (a) $t = 2\tau_d$ and (d) $t = 10\tau_d$, the case where the polynomial $\hat{p}_d$ has been computed to approximate the f in the interval $[-1, -a]$. Dashed lines indicate the error bound of Equation (22). The initial condition is the same employed as Figure 8, represented as $t = 0$. Parameters are set to $N = 64$ sites, $\frac{D}{\Delta x^2} = 0.2$, $a = 0.1$, with varying polynomial degree for QSVT approximating the even function $f(x) = e^{-t|x|}$.

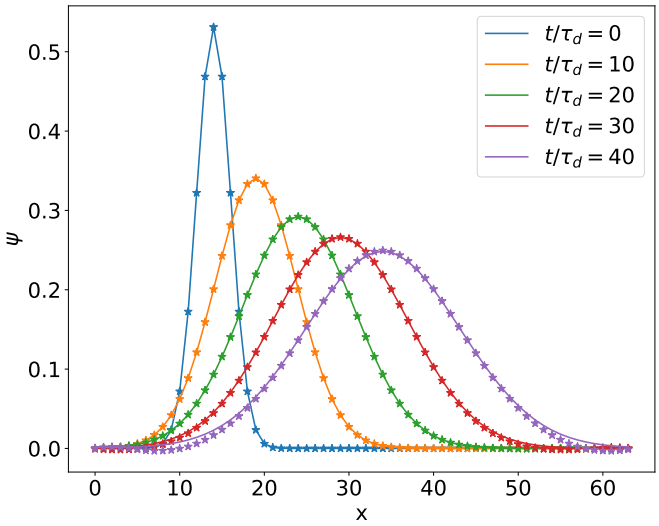


Figure 8: Comparison between the exact solution (solid line) and the QSVT simulation (markers) setting $N = 64$ sites, $\frac{D}{\Delta x^2} = 0.2$, $a = 0.1$, $c = 0.1$, 101 angles for QSVT approximating the even function $f(x) = e^{-t|x|}$ with a degree $d = 100$ polynomial expansion targeted to the interval $[-1, -a]$. Time is expressed in expressed in units of the cell diffusion time $\tau_d = \Delta x^2/D$. The initial condition is a Gaussian function centered in the site 15, with a standard deviation of 2.

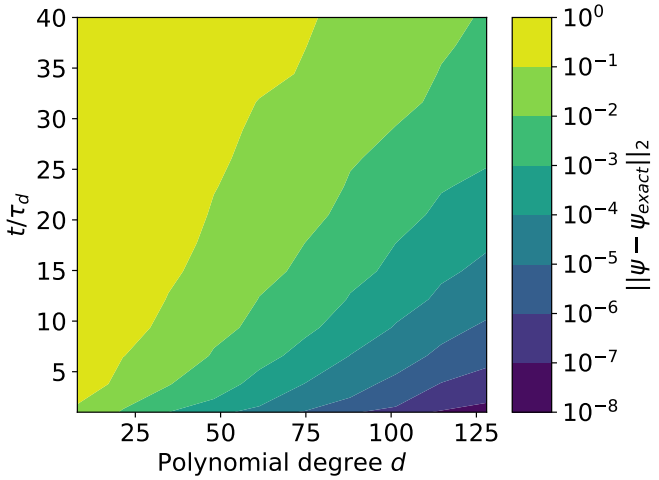


Figure 9: Error in the Euclidean norm as a function of the polynomial degree d and the simulation time t, expressed in units of the cell diffusion time $\tau_d = \Delta x^2/D$. The polynomial approximation is targeted to the interval $[-1, -a]$.

tion is computed numerically. Therefore, the inherent simplicity of the ADR equation makes it an ideal testbed for assessing the feasibility of a quantum simulation of classical dynamics.

In this work, we have proposed and fully characterized a quantum algorithm for simulating one-dimensional ADR dynamics within the QSVT framework. QSVT is a powerful tool capable of implementing approximate propagators for dynamical systems. Specifically, we used QSVT to derive a telescopic solution for long-time dynamics, thereby circumventing the time-discretization techniques, while directly solving the original differential equation. Thus, we have extensively studied the quantum circuit along with the numerical methods employed to achieve the correct transformation that accurately represents the state at the final simulation time by employing a polynomial approximation over a restricted interval of the exponential function. We have analyzed the dependence of the simulation error on the degree of the polynomial approximation together with the temporal range of the telescopic simulation. Polynomial degree and temporal range are related together, as longer simulation times need a higher polynomial degree to achieve an acceptable error and vice versa. The numerical results show that using polynomial expansion of degree 100 provides a low error for a simulation time comparable to 40 cell diffusion times, but so far reaching the time scale $\tau_{diff} = L^2/D = (2^n\Delta x)^2/D$ is beyond the capability of the proposed method as the number of qubits n increases. Furthermore, we have derived an analytical upper bound for the Euclidean error of the simulation, with the twofold goal of verifying the reliability of the polynomial expansion, and identify the necessary expansion degree to maintain the error within a specified threshold. Hence, we have fully analyzed the quantum circuit. For the constant coefficient case an efficient block-encoding technique allows to implement the circuit with a reduced number of ancilla qubits. Moreover, we have found that the depth of the circuit scales as $n^{3.7}$, a polynomial growth that remains within the capabilities of near terms quantum hardware [36] for a relevant number of spatial lattice sites.

Despite the promising results, we emphasize that our findings pertain an extremely simple scenario. A more difficult challenge that requires further investigation is to tackle the ADR dynamics in the presence of non constant advection, diffusion and reaction coefficients, for which an analytical solution is not known. To perform a quantum simulation of this scenario, one should find an efficient block encoding of the discretized differential equation. Once such encoding is realized, QSVT could be applied to the block-encoded matrix, and our analysis of the error would remain pertinent. Another application where QSVT offers a potential computational advantage is the simulation of nonlinear systems in even more complex scenarios, like the fluid dynamics described by Navier-Stokes equations for time-dependent incompressible flows. Similarly, the QSVT framework could be applied to the Carleman lattice Boltzmann method [9, 37], which can provide a good degree of approximation to the solution of the Navier-Stokes equations, to obtain the result at the final simulation time with an appropriate polynomial transformation.

References

[1] M Sanz et al. "Quantum simulator for transport phenomena in fluid flows". In: *Scientific reports* 5.1 (2015), p. 13153. DOI: https://doi.org/10.1038/srep13153.

[2] Sauro Succi et al. "Quantum computing for fluids: Where do we stand?" en. In: *Europhysics Letters* 144.1 (Oct. 2023), p. 10001. ISSN: 0295-5075, 1286-4854. DOI: 10.1209/0295-5075/acfdc7.

[3] Xiangyu Li et al. "Potential quantum advantage for simulation of fluid dynamics". In: *Phys. Rev. Res.* 7 (1 Jan. 2025), p. 013036. DOI: 10.1103/PhysRevResearch.7.013036.

[4] Antonio David Bastida Zamora et al. "Efficient quantum lattice gas automata". In: *Computers & Fluids* 286 (2025), p. 106476. ISSN: 0045-7930. DOI: https://doi.org/10.1016/j.compfluid.2024.106476.

[5] Peter Love. "On Quantum Extensions of Hydrodynamic Lattice Gas Automata". In: *Condensed Matter* 4.2 (2019). ISSN: 2410-3896. DOI: 10.3390/condmat4020048.

[6] NiccolÃ² Fonio, Pierre Sagaut, and Giuseppe Di Molfetta. "Quantum collision circuit, quantum invariants and quantum phase estimation procedure for fluid dynamic lattice gas automata". In: *Computers & Fluids* 299 (2025), p. 106688. ISSN: 0045-7930. DOI: https://doi.org/10.1016/j.compfluid.2025.106688.

[7] Blaga N. Todorova and RenÃ© Steijl. "Quantum algorithm for the collisionless Boltzmann equation". In: *Journal of Computational Physics* 409 (2020), p. 109347. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2020.109347.

[8] Merel A. Schalkers and Matthias Möller. "Momentum exchange method for quantum Boltzmann methods". In: *Computers & Fluids* 285 (2024), p. 106453. ISSN: 0045-7930. DOI: https://doi.org/10.1016/j.compfluid.2024.106453.

[9] Claudio Sanavio and Sauro Succi. "Lattice Boltzmann–Carleman quantum algorithm and circuit for fluid flows at moderate Reynolds number". In: *AVS Quantum Science* 6.2 (2024). DOI: https://doi.org/10.1116/5.0195549.

[10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. DOI: https://doi.org/10.1017/CBO9780511976667.

[11] András Gilyén et al. "Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics". en. In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. Phoenix AZ USA: ACM, June 2019, pp. 193–204. ISBN: 978-1-4503-6705-9. DOI: 10.1145/3313276.3316366.

[12] I. Novikau, I.Y. Dodin, and E.A. Startsev. "Simulation of Linear Non-Hermitian Boundary-Value Problems with Quantum Singular-Value Transformation". In: *Phys. Rev. Appl.* 19 (5 May 2023), p. 054012. DOI: 10.1103/PhysRevApplied.19.054012.

[13] Alexis Ralli et al. *Calculating the Single-Particle Many-body Green's Functions via the Quantum Singular Value Transform Algorithm*. en. arXiv:2307.13583 [quant-ph]. July 2023. DOI: 10.48550/arXiv.2307.13583.

[14] Ivan Novikau and Ilon Joseph. "Estimating QSVT angles for matrix inversion with large condition numbers". In: *Journal of Computational Physics* 525 (2025), p. 113767. DOI: https://doi.org/10.1016/j.jcp.2025.113767.

[15] John M. Martyn et al. "Grand Unification of Quantum Algorithms". en. In: *PRX Quantum* 2.4 (Dec. 2021), p. 040203. ISSN: 2691-3399. DOI: 10.1103/PRXQuantum.2.040203.

[16] Chi-Fang Chen et al. *Quantum Thermal State Preparation*. 2023. DOI: https://doi.org/10.48550/arXiv.2303.18224. arXiv: 2303.18224 [quant-ph].

[17] Richard P Feynman. "Simulating physics with computers". In: *International journal of theoretical physics* 21.6/7 (1982), pp. 467–488. DOI: https://doi.org/10.1007/BF02650179.

[18] Guang Hao Low, Theodore J. Yoder, and Isaac L. Chuang. "Methodology of Resonant Equiangular Composite Quantum Gates". In: *Physical Review X* 6.4 (Dec. 2016). ISSN: 2160-3308. DOI: 10.1103/physrevx.6.041067.

[19] Guang Hao Low and Isaac L Chuang. "Hamiltonian simulation by qubitization". In: *Quantum* 3 (2019), p. 163. DOI: https://doi.org/10.22331/q-2019-07-12-163.

[20] Lov K Grover. "A fast quantum mechanical algorithm for database search". In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996, pp. 212–219. DOI: https://doi.org/10.1145/237814.237866.

[21] Mario Szegedy. *Spectra of Quantized Walks and a $\sqrt{\delta\epsilon}$ rule*. 2004. DOI: https://doi.org/10.48550/arXiv.quant-ph/0401053. arXiv: quant-ph/0401053 [quant-ph].

[22] Daan Camps et al. "Explicit quantum circuits for block encodings of certain sparse matrices". In: *SIAM Journal on Matrix Analysis and Applications* 45.1 (2024), pp. 801–827. DOI: https://doi.org/10.1137/22M1484298.

[23] Leigh Lapworth. *Evaluation of block encoding for sparse matrix inversion using QSVT*. 2024. DOI: https://doi.org/10.48550/arXiv.2402.17529. arXiv: 2402.17529 [quant-ph].

[24] Yulong Dong et al. "Efficient phase-factor evaluation in quantum signal processing". In: *Physical Review A* 103.4 (Apr. 2021). Publisher: American Physical Society, p. 042419. DOI: 10.1103/PhysRevA.103.042419.

[25] Yulong Dong et al. "Robust iterative method for symmetric quantum signal processing in all parameter regimes". In: *SIAM Journal on Scientific Computing* 46.5 (2024), A2951–A2971. DOI: https://doi.org/10.1137/23M1598192.

[26] Lin Lin. *Lecture Notes on Quantum Algorithms for Scientific Computation*. 2022. DOI: https://doi.org/10.48550/arXiv.2201.08309. arXiv: 2201.08309 [quant-ph].

[27] Claudio Sanavio, Enea Mauri, and Sauro Succi. "Explicit Quantum Circuit for Simulating the Advection–Diffusion–Reaction Dynamics". In: *IEEE Transactions on Quantum Engineering* 6 (2025), pp. 1–12. DOI: 10.1109/TQE.2025.3544839.

[28] Alessandro Andrea Zecchi et al. "Improved amplitude amplification strategies for the quantum simulation of classical transport problems". In: *Quantum Science and Technology* 10.3 (June 2025), p. 035039. DOI: 10.1088/2058-9565/addeea.

[29] Lloyd N Trefethen. *Approximation theory and approximation practice, extended edition*. SIAM, 2019. DOI: https://doi.org/10.1137/1.9781611975949.

[30] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical mathematics*. Vol. 37. Springer Science & Business Media, 2010. DOI: https://doi.org/10.1007/978-0-387-22750-4.

[31] Evgeny Y Remez. "Sur le calcul effectif des polynomes d'approximation de Tschebyscheff". In: *CR Acad. Sci. Paris* 199.2 (1934), pp. 337–340.

[32] Steven Diamond and Stephen Boyd. "CVXPY: A Python-embedded modeling language for convex optimization". In: *Journal of Machine Learning Research* 17.83 (2016), pp. 1–5.

[33] Akshay Agrawal et al. "A rewriting system for convex optimization problems". In: *Journal of Control and Decision* 5.1 (2018), pp. 42–60. DOI: https://doi.org/10.1080/23307706.2017.1397554.

[34] Kazuki Sakamoto and Keisuke Fujii. "On the quantum computational complexity of classical linear dynamics with geometrically local interactions: Dequantization and universality". In: *arXiv preprint arXiv:2505.10445* (2025). DOI: https://doi.org/10.48550/arXiv.2505.10445.

[35] Ali Javadi-Abhari et al. *Quantum computing with Qiskit*. 2024. DOI: 10.48550/arXiv.2405.08810. arXiv: 2405.08810 [quant-ph].

[36] Vincent R Pascuzzi and Antonio Córcoles. "Quantum-centric Supercomputing for Physics Research". In: *arXiv preprint arXiv:2408.11741* (2024). DOI: https://doi.org/10.48550/arXiv.2408.11741.

[37] Claudio Sanavio et al. "Three Carleman routes to the quantum simulation of classical fluids". In: *Physics of Fluids* 36.5 (2024). DOI: https://doi.org/10.1063/5.0204955.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

52/2025 Zecchi, A. A.; Sanavio, C.; Perotto, S.; Succi, S.
Telescopic quantum simulation of the advection-diffusion-reaction dynamics

51/2025 Tomasetto, M.; Braghin, F., Manzoni, A.
Latent feedback control of distributed systems in multiple scenarios through deep learning-based reduced order models

50/2025 Bonetti, S.; Botti, M.; Antonietti, P.F.
Conforming and discontinuous discretizations of non-isothermal Darcy–Forchheimer flows

49/2025 Zanin, A.; Pagani, S.; Corti, M.; Crepaldi, V.; Di Fede, G.; Antonietti, P.F.; the ADNI
Predicting Alzheimer's Disease Progression from Sparse Multimodal Data by NeuralODE Models

48/2025 Temellini, E.; Ballarin, F.; Chacon Rebollo, T.; Perotto, S.
On the inf-sup condition for Hierarchical Model reduction of the Stokes problem

47/2025 Gimenez Zapiola, A.; Consolo, A.; Amaldi, E.; Vantini, S.
Penalised Optimal Soft Trees for Functional Data

46/2025 Mirabella, S.; David, E.; Antona, A.; Stanghellini, C.; Ferro, N.; Matteucci, M.; Heuvelink, E.; Perotto, S.
On the Impact of Light Spectrum on Lettuce Biophysics: A Dynamic Growth Model for Vertical Farming

45/2025 Caliò, G.; Ragazzi, F.; Popoli, A.; Cristofolini, A.; Valdettaro, L; De Falco, C.; Barbante, F.
Hierarchical Multiscale Modeling of Positive Corona Discharges

44/2025 Brivio, S.; Fresca, S.; Manzoni, A.
Handling geometrical variability in nonlinear reduced order modeling through Continuous Geometry-Aware DL-ROM

43/2025 Tomasetto, M.; Manzoni, A.; Braghin, F.
Real-time optimal control of high-dimensional parametrized systems by deep-learning based reduced order models