Turbulent Mixing Simulation via a Quantum Algorithm

Guanglei Xu[a] and Andrew J. Daley[b]
University of Strathclyde, Glasgow G4 0NG, United Kingdom.

Peyman Givi[c]
University of Pittsburgh, Pittsburgh, PA 15261, USA.

Rolando D. Somma[d]
Los Alamos National Laboratory, Los Alamos, NM 87545, USA.

(Dated: July 30, 2017)

Probability density function (PDF) methods have been very useful in describing many physical aspects of turbulent mixing. In applications of these methods, modeled PDF transport equations are commonly simulated via classical Monte Carlo techniques, which provide estimates of moments of the PDF at arbitrary accuracy. In this work, we use recently developed techniques in quantum computing and quantum enhanced measurements (quantum metrology) to construct a quantum algorithm that accelerates the computation of such estimates. Our quantum algorithm provides a quadratic speedup over classical Monte Carlo methods in terms of the number of repetitions needed to achieve the desired precision. We illustrate the power of our algorithm by considering a binary scalar mixing process modeled by means of the coalescence/dispersion (C/D) closure. The equation is first simulated using classical Monte Carlo methods, where we provide error estimates for the computation of central moments. We then simulate the quantum algorithm for this problem by sampling from the same probability distribution as that of the output of a quantum computer, and show that significantly less resources are required to achieve the same precision.

[a] Ph.D. Candidate, Department of Physics and SUPA.
[b] Professor, Department of Physics and SUPA.
[c] Professor, Mechanical Engineering and Petroleum Engineering, AIAA Fellow.
[d] Staff member, Theoretical Division.

Our results demonstrate potential applications of future quantum computers for simulation of turbulent mixing, and large classes of related problems.

Nomenclature

α = the random variable that determines the conditions of mixing

$\mathcal{D}$ = the conditional expected value of the scalar diffusion

$\mathcal{E}$ = the conditional expected value of the scalar dissipation

${}^{c}U$ = a controlled unitary transformation matrix

ϵ = the precision of estimation

ϵ_C = the statistical error of classical Monte Carlo methods

ϵ_Q = the estimation error of quantum algorithm

Γ = the binary Fickian diffusion coefficient

$\hat{\mu}_l$ = the estimator of the l-th central moment of probability distribution

$\hat{\mu}_l^k(t_j)$ = the estimator of the l-th central moment of distribution of k-th realization, at time t_j

$\hat{\sigma}_{\mu_4}$ = the estimated standard deviation associated with $\hat{\mu}_4$

$|0\rangle$, $|1\rangle$ = eigenstates of computational basis of one qubit

$|l\rangle$ = a state in the computational basis, which l is the corresponding binary representation

$\mu_l(t)$ = the l-th central moment of probability distribution at time t

ω = the mixing frequency of binary scalar mixing problem

$\phi(\underline{\mathrm{x}}, t)$ = a Fickian scalar, where $\underline{\mathrm{x}}$ is the position vector and $t \geq 0$ represents time

ϕ_l, ϕ_u = the lower bound and upper bound of Fickian scalars

ψ = the composition domain of Fickian scalars ϕ

σ^2 = the variance of random variables

σ_i = the Pauli matrices, $i = 0, x, y, z$

$\tilde{\mu}_4$ = the most accurate estimate of μ_4 obtained, used as a normalization parameter in some figures

$A(\alpha)$ = the probability density function of the random variable α

a_m = the m-th moments of the random variable α

b_i' = the measurement outcome of i-th qubit in phase estimation algorithm

c = the confidence level of estimation

N_p = the total number of simulation particles in classical Monte Carlo method

N_r = the total number of repetitions

N_s = the total number of mixing within one time step

N_t = the total number of time steps

$P(\psi, t)$ = the single-point probability density function of the scalar ψ at time t

$R(\nu, \theta)$ = unitary rotation matrices along certain axis on Bloch sphere, $\nu = x, y, z$

BPP = bounded-error probabilistic polynomial time, a class of problems in computational complexity theory

BQP = bounded-error quantum polynomial time, a class of problems in computational complexity theory

CNOT = Controlled-NOT gate

I. Introduction

Large quantum computers could provide answers to problems that are believed to be otherwise intractable (c.f. [1]). Particularly, in recent years, there has been significant interest in the development of quantum algorithms to speed up classical Monte Carlo (MC) techniques [2–10]. As MC techniques are used ubiquitously in science, the existence of large-scale quantum computers has the potential to revolutionize computation across a wide range of disciplines. As hardware for quantum computing undergoes continued rapid development [11–18], an important step is to ask what impact such new methods might have in different disciplines, and to identify specific examples where there can be a large impact of this emerging technology. Here we aim to bring together research in quantum computing with fluids engineering, by identifying a general class of problems relevant to turbulent flows that can be sped up on a quantum computer. This class of algorithms can act as a starting point for further algorithmic development, and to begin to answer detailed questions about the quantum hardware necessary to run algorithms for aerospace applications.

In fluid mechanics, classical MC methods have been widely used for turbulence simulation; in particular for description of turbulent scalar mixing (with or without chemical reactions). Understanding the mixing phenomenon has been a subject of broad interest for the past fifty years in various disciplines of engineering [19–27]. The underlying basic physics is explicitly captured by probability density function (PDF) methods in the contexts of both Reynolds averaged Navier-Stokes (RANS) [28, 29] and large eddy simulation (LES) [30, 31]. In the setting of a spatially homogeneous flow, the temporal evolution of the scalar PDF isolates the physical features pertaining to mixing transport. In this setting, in addition to the accuracy of its closure, the computational efficiency of the PDF simulator is of significant importance.

Classical MC methods have been the primary means of solving PDF transport equations [32–34]. With these methods, the PDF is represented by an ensemble of computational elements or particles. The transport and the changes in the composition of these particles are made randomly in a such a way as to mimic the (modeled) physics of the problem. The ensemble average of data over these particles then determines the desired statistics. To obtain accurate results, a MC method needs to be executed repeatedly many times. The complexity of MC depends on various parameters,

including the desired final precision $\epsilon \ll 1$ and the confidence level of the estimation, c. When the quantity to be estimated by the MC method has bounded moments, as is the case in turbulent mixing, we can use Chebyshev's inequality to obtain N_r, the number of MC runs needed to obtain such an estimate. It is well known that N_r is of order σ^2/ϵ^2, where σ^2 is the variance of the random variable. (The dependence of N_r on the confidence level is only logarithmic in $(1-c)^{-1}$.) The complexity dependence of MC on $1/\epsilon^2$ is undesired and may not be avoided by using other conventional techniques. Novel algorithms that have a better complexity dependence on $1/\epsilon$ are thus highly desirable.

Remarkably, quantum computers would allow us to achieve a quadratic complexity improvement over the classical bound for certain problems. Recent results in "quantum metrology" (i.e., quantum enhanced measurements) [5, 6, 35] demonstrate that quantum computers can provide a quadratic improvement in the precision of certain estimations using the same number of resources as classical computers. Equivalently, for the same precision, quantum computers would require quadratically less resources than classical ones in these cases. These results are somewhat general in that they allow us to estimate expected values of various quantities under minimal assumptions and within an arbitrary confidence level [6]. The complexity overhead to achieve confidence c is also logarithmic in $(1-c)^{-1}$. Only recently these quantum-metrology based methods have been adapted to improve upon the complexity of classical MC methods (c.f., [9]), but their potential application to specific cases and their usefulness in those instances have not yet been investigated.

Here, we adapt the quantum metrology results of [6] to the setting of turbulent mixing flows and present a quantum algorithm that is quadratically more efficient, in terms of the number of repetitions, than MC methods. Our quantum algorithm can be used to simulate large classes of turbulent mixing problems including those modeled by means of the coalescence/dispersion (C/D) closure [36–38]. In more detail, we provide a quantum algorithm to compute properties of the PDF. The precision of the estimation, ϵ, depends almost linearly on $1/N_r$. Here, N_r refers to the number of times a certain quantum state has to be prepared and can be compared to the number of times a MC method is executed. Equivalently, for target precision ϵ, N_r has to be chosen to be of order $1/\epsilon$. Thus, the quantum algorithm provides a quadratic speedup over MC in terms of N_r to achieve

the same precision.

We demonstrate a specific application of our quantum algorithm by considering a simple binary scalar mixing process modeled by the C/D closure. While an analytical solution for the scalar moments is possible in this case, our simulations illustrate what is possible and allow a quantitative analysis of the corresponding statistical errors. The algorithm can then be used to attack more general mixing processes. To understand the complexity of the classical algorithm, binary mixing is then first simulated using classical MC methods, where we provide estimates and error bars for the 4-th central moment of the PDF (the calculation of higher order moments can be performed similarly). We then simulate the quantum algorithm using conventional techniques (as large quantum computers do not yet exist) by sampling from the same probability distribution as that of the measurement outcome of the quantum computer. Note that it would be impossible to simulate the full quantum algorithm as the number of qubits (quantum bits) needed would be very large and conventional simulations of quantum algorithms would require dealing with matrices that are of dimension exponential in the number of qubits. This would limit conventional simulations of quantum algorithms to about 40 qubits using supercomputers, and our quantum algorithms require significantly many more qubits to be implemented. This reflects the clear differences between a quantum computer, and classical computers. Nevertheless, the probability distribution associated with the output of the quantum algorithm can be obtained precisely in this case due to the simplicity of the problem, but this would not be possible for a more general case. The results of the simulation of the quantum algorithm clearly show significantly (quadratically) smaller error bars for the estimation of the 4-th central moment using the same value for N_r. A similar result would hold for the computation of other properties of the PDF.

For simplicity, our quantum algorithms are described in a sequence of steps, each to estimate different significant bits of the quantity of interest. It follows that the depth of the quantum circuits is of order $1/\epsilon$. This seems to be a drawback with respect to classical MC methods that can be trivially parallelized. However, we show that our quantum algorithms can also be parallelized, resulting in quantum circuits of relatively short depth, and where the total number of qubits required is linear in $1/\epsilon$.

This paper is organized as follows. In Sec. II we provide an introduction to turbulent scalar mixing problems and describe the C/D closure. In Sec. III we discuss classical MC methods for turbulent mixing, and taking a C/D model as an example, we implement MC to simulate binary mixing as a demonstration. Previous to describing our quantum algorithm, in Sec. IV we provide a brief overview of the required background concepts in quantum information. Then, in Sec. V, we present our quantum algorithm to simulate the C/D model and provide the simulation results of binary mixing to understand the advantages of the quantum algorithm with respect to MC. A procedure for parallelizing the quantum algorithm is also presented in Sec. V. We finish with a conclusion and outlook in Sec. VI.

II. Turbulent scalar mixing

In this section, we introduce the basic problem of turbulent scalar mixing by means of the single-point PDF transport equation. In particular, we consider a homogeneous turbulent flow in which the PDF closure problem is exhibited and present one model for the closure. We consider the mixing of a Fickian scalar $\phi = \phi(\underline{\mathrm{x}}, t)$, where $\underline{\mathrm{x}}$ is the position vector and $t \geq 0$ denotes time; from an initially binary state within bounds $\phi_\ell \leq \phi \leq \phi_u$. The PDF of the scalar is $P(\psi, t)$, where ψ is the composition domain of ϕ. In homogeneous turbulent flows, where statistics are spatially invariant, $P(\psi, t)$ is governed by either of the two equations [26]:

$$\frac{\partial P(\psi,t)}{\partial t} + \frac{\partial^2 \left(\mathcal{E} P(\psi,t)\right)}{\partial \psi^2} = 0 \,, \tag{1}$$

$$\frac{\partial P(\psi,t)}{\partial t} + \frac{\partial (\mathcal{D} P(\psi,t))}{\partial \psi} = 0 \,. \tag{2}$$

Here, $\mathcal{E}$ represents the expected value of the scalar dissipation conditioned on the scalar value $\phi(\underline{\mathrm{x}}, t)$ and $\mathcal{D}$ denotes the conditional expected value of the scalar diffusion:

$$\mathcal{E}(\psi,t) = E[\Gamma \nabla\phi \cdot \nabla\phi | \phi(\underline{\mathrm{x}},t) = \psi], \quad \mathcal{D}(\psi,t) = E[\Gamma \nabla^2 \phi | \phi(\underline{\mathrm{x}},t) = \psi], \tag{3}$$

where Γ is the binary Fickian diffusion coefficient. We use the standard notation where $E[y]$ and $E[y|z]$ denote the expected value of a random variable y and the expected value conditioned on an event z, respectively. With the single-point statistical descriptor $P(\psi, t)$, the turbulence closure problem is exhibited by the unknown conditional / unconditional dissipation, and/or the conditional

diffusion. A variety of models have been proposed and employed for the PDF closure [36–47]. This remains as an area of active investigation and the search continues for a model that satisfies various mixing scenarios [48]. The available models are either written in terms of a Langevin equation with the corresponding Fokker-Planck equation describing the PDF, or via a phenomenological transport equation for the PDF evolution [49].

For the purpose of demonstration, here we consider the family of coalescence/dispersion (C/D) mixing models. The generalized C/D model is described by the evolution equation [37, 38]

$$\frac{\partial P(\psi,t)}{\partial t} = -2\beta\omega P(\psi,t) + 2\beta\omega \int d\psi' \int d\psi'' P(\psi',t)P(\psi'',t) \int_0^1 d\alpha A(\alpha)\delta[\psi - (1-\alpha)\psi' - \frac{1}{2}\alpha(\psi'+\psi'')] \,, \quad (4)$$

where $\delta(x)$ is the Dirac delta function, and $A(\alpha)$ is the PDF of the random variable α, $0 \leq \alpha \leq 1$. The value of α determines the conditions of mixing. In particular, to obtain Curl's model [50], we choose $A(\alpha) = \delta(\alpha - 1)$; for the closure of Janicka *et al.* [36], $A(\alpha) = 1$; and for the least mean square estimation (LMSE) [28, 39], and the interaction by exchange with the mean (IEM) model [51], $A(\alpha) = \delta(\alpha - \zeta)$, with $\zeta \to 0$. The parameter ω is the mixing frequency and determines the rate of variance decay. The parameter β depends on $A(\alpha)$ as follows:

$$\beta = \frac{1}{a_1 - \frac{1}{2}a_2} \,, \; a_m = \int_0^1 d\alpha \; \alpha^m A\left(\alpha\right) \; . \quad (5)$$

In this way, all C/D models have the same rate of variance decay.

The properties of the PDF can be described by the central moments, which are defined via ($l = 1, 2, \ldots$)

$$\mu_l\left(t\right) = E[(\psi - E[\psi])^l] \; . \quad (6)$$

In certain cases, these moments can be obtained exactly – which will be useful here in demonstrating the accuracy obtained by our algorithms. For the problem of binary scalar mixing, we take $P(\psi, t = 0) = \frac{1}{2}[\delta(\psi - \phi_\ell) + \delta(\psi - \phi_u)]$, and use Curl's model with bounds $\phi_\ell = -1$, $\phi_u = 1$. The central

moments can then be obtained exactly as:

$$\mu_1(t) = \mu_1(0) = 0 \,, \tag{7}$$

$$\mu_2\,(t) \equiv \sigma^2\,(t) = e^{-2\omega t} \,, \tag{8}$$

$$\mu_3\,(t) = \mu_3(0) = 0, \tag{9}$$

$$\mu_4\,(t) = \left(4e^{\gamma\omega t} - 3\right) e^{-4\omega t} \,, \tag{10}$$

where

$$\gamma = \frac{a_2 + \frac{1}{4}a_4 - a_3}{a_1 - \frac{1}{2}a_2} \,, \tag{11}$$

and a_m are the m-th moments of the random variable α.

III. Monte Carlo methods for the C/D model

To simulate the C/D model [Eq. (4)] via a classical MC method, one chooses a number of "particles" N_p so that each particle has an associated random variable $\psi^k(i, t_j)$, where $i = 1, \ldots, N_p$, $j = 0, \ldots, N_t$, and $k = 1, \ldots, N_r$. These particles are intended to simulate the different populations of ψ. The variable t_j refers to the time at the j-th step of the algorithm in any run, and $t_j = j\Delta t$, for some $\Delta t > 0$. The algorithm is repeated N_r times to reach a desired accuracy. The total evolution time $t > 0$, together with β and ω, are parameters defined by the physical properties of the system (Sec. II). The parameters Δt, N_p, and N_r are "experimentally" determined depending on the desired accuracy of the simulation and computing resources, or can be considered as inputs. The classical MC algorithm is described in detailed as follows:

Input: $P(\psi,0)$**,** t**,** β**,** ω**,** Δt**,** N_p**,** N_r

1. Obtain $N_t = \lceil t/\Delta t \rceil$, $N_s = \lceil \beta\omega\Delta t N_p \rceil$. Set $k = 1$, $j = 1$, $n_s = 1$, and $t_0 = 0$.

2. Repeat until $k > N_r$:

2.1. For $i = 1, \ldots, N_p$, initialize $\psi^k(i,0)$ according to an initial probability distribution $Q(\psi^k(1,0), \ldots, \psi^k(N_p,0))$.

2.2. Repeat until $j > N_t$:

2.2.1. Set $t_j = j\Delta t$ and $\psi^k(i,t_j) := \psi^k(i,t_{j-1})$ for all $i \in \{1, \ldots, N_p\}$.

2.2.2. Repeat until $n_s > N_s$:

2.2.2.1. Obtain random integers $i_1, i_2 \in \{1, \ldots, N_p\}$.

2.2.2.2. Sample $\alpha \in [0,1]$ according to the probability distribution $A(\alpha)$.

2.2.2.3. Perform the mixing transformation:

$$\psi^k(i_1,t_j) \leftarrow (1-\alpha)\psi^k(i_1,t_j) + \alpha(\psi^k(i_1,t_j) + \psi^k(i_2,t_j))/2,$$
$$\psi^k(i_2,t_j) \leftarrow (1-\alpha)\psi^k(i_2,t_j) + \alpha(\psi^k(i_1,t_j) + \psi^k(i_2,t_j))/2.$$

2.2.2.4. $n_s \leftarrow n_s + 1$.

2.2.3. $j \leftarrow j + 1$.

2.3. $k \leftarrow k + 1$:

3. **Output**: $\psi^k(i,t_j)$ for all k, i, t_j.

The initial distribution $Q(\psi^k(1,0), \ldots, \psi^k(N_p,0))$ is independent of k and is chosen so that it simulates $P(\psi,0)$ in the PDF transport equation. At any time, the distribution associated with the MC method is $Q(\psi^k(1,t_j), \ldots, \psi^k(N_p,t_j))$. Then, the results of the MC method are used to obtain an estimate of $P(\psi,t)$ or estimate quantities such as the l-th central moment of ψ. To obtain the simulated PDF, one technique is to build a histogram with the values of each $\psi^k(i,t_j)$ for $i = 1, \ldots, N_p$, and then choose a corresponding (machine) precision $\Delta\psi$ and use a proper

normalization. Each MC run outputs a random vector $(\psi^k(1,t),\ldots,\psi^k(N_p,t))$ that is independent for each k but its entries may be (slightly) correlated for each k.

The MC method can then be used to estimate different central moments of the distribution $P(\psi,t)$ as follows:

$$\hat{\mu}_l(t_j) := \frac{1}{N_r}\sum_{k=1}^{N_r}\hat{\mu}_l^k(t_j)\ , \tag{12}$$

with

$$\hat{\mu}_l^k(t_j) := \frac{1}{N_p}\sum_{i=1}^{N_p}\left(\psi^k(i,t_j)-\hat{E}[\psi^k(t_j)]\right)^l\ . \tag{13}$$

We use the standard notation where $\hat{X}$ denotes an estimator of X; in this case, $\hat{E}[\psi^k(t_j)]$ is the estimator of the expected value of $\psi^k(i,t_j)$:

$$\hat{E}[\psi^k(t_j)] := \frac{1}{N_p}\sum_{i=1}^{N_p}\psi^k(i,t_j)\ . \tag{14}$$

As described, the MC algorithm takes N_r as input and outputs all the $\psi^k(i,t_j)$. Straightforward modifications of the algorithm would take a precision parameter ϵ as input and would output certain properties of the PDF, such as the central moments with the corresponding error bounds and confidence levels, rather than keeping all values of $\psi^k(i,t_j)$ in memory. Such algorithmic modifications may render the algorithm more efficient.

A. Complexity

We study the complexity of the previous MC method. For simplicity, we disregard certain logarithmic factors in the order notation. Disregarding the complexity of initializing $\psi^k(1,0),\ldots,\psi^k(N_p,0)$ and the complexity of sampling from $A(\alpha)$, the complexity of the MC method is mainly dominated by the number of times the $\psi^k(i,t_j)$ have to be updated. This is $O(N_r\ N_t\ N_s) = O(N_r t\beta\omega N_p)$. Both N_r and N_p are set to reach desired accuracy in the computation of the relevant quantities. Under the assumptions on ψ, for fixed N_p and fixed confidence level c, the overall precision ϵ is then dependent on the number of repetitions N_r. Chebyshev's inequality implies $\epsilon = O(1/\sqrt{N_r})$. For arbitrary values of $c<1$, the overhead is only logarithmic in $(1-c)^{-1}$ [52]. When considering ϵ as an input, the complexity of the MC method is $O(t\beta\omega N_p/\epsilon^2)$.

B. Example: Classical Monte Carlo simulations of binary mixing

To demonstrate the MC method and to give us a basis for comparison with our quantum algorithm, we simulate a simple binary mixing problem whose solution can be analytically obtained. This will facilitate the benchmarking of our algorithms and how they might be expected to perform when applied to a complex problem where the solution is unknown. We consider Curl's model, where $A(\alpha) = \delta(\alpha - 1)$, and $\beta = 2$, $\gamma = 0.5$. The mixing frequency is set to $\omega = 1$. The maximum simulation time is $t = 1$ and the other parameters are set to $\Delta t = 0.1$ and $N_p = 10^3$. The initial PDF is the binary state $P(\psi, 0) = \frac{1}{2}\delta(\psi - 1) + \frac{1}{2}\delta(\psi + 1)$, and we simulate it by initially setting $\psi^k(i, 0) = -1$ for all $1 \leq i \leq N_p/2$, and $\psi^k(i, 0) = +1$ otherwise. We can then use the MC method to estimate the 4-th central moment as a function of time, as described in Sec. III. Since the known analytical solution refers to the case where $N_p = \infty$ and we need the solution for $N_p < \infty$, we perform a very accurate simulation for $N_p = 10^3$ by repeating the MC method $N_r = 2^{20} \times 60$ times. (The reason why we factor the coefficients in N_r will become clear when we discuss the quantum algorithm.) The 4-th central moment of such an accurate estimate is $\tilde{\mu}_4(t)$ and is obtained from the simulation results using Eq. (12). We then use $\hat{\mu}_4(t)$ to denote the estimated 4-th central moment for other smaller values of N_r, also obtained via Eq. (12).

The MC results are shown in Fig. 1. In Fig. 1 (a) we show the exponential decay $\hat{\mu}_4(t)$ as a function of time for $N_r = 2^{10} \times 24$. In Fig. 1 (b) we compare $\hat{\mu}_4(t)$ with $\tilde{\mu}_4(t)$, which is very close to the exact solution for $N_p = 10^3$. Note that $\hat{E}[\psi^k(t_j)] = 0$ in this case [see Eq. (14)]. To obtain the error bars of Fig. 1 (b), we first computed $\hat{\mu}_4^k(t_j)$ for each run $k = 1, \ldots, N_r$, according to Eq. (13). Then, the estimated standard deviation associated with $\hat{\mu}_4(t)$ is

$$\hat{\sigma}_{\mu_4}(t_j) = \left[\frac{\sum_{k=1}^{N_r} (\hat{\mu}_4^k(t_j) - \hat{\mu}_4(t_j))^2}{(N_r - 1)} \right]^{1/2} . \tag{15}$$

An estimate of $\mu_4(t)$ within $\epsilon_C(t_j) = 3\hat{\sigma}_{\mu_4}(t_j)$ allows us to reach 99.75% confidence level. The error bars of Fig. 1 (b) denote the regions

$$\left[\frac{\hat{\mu}_4(t_j) - \epsilon_C(t_j)}{\tilde{\mu}_4(t_j)}, \frac{\hat{\mu}_4(t_j) + \epsilon_C(t_j)}{\tilde{\mu}_4(t_j)} \right] . \tag{16}$$

From the simulation results and the analysis above, we observe that the dependence of the estimation errors on the number of repetitions is of order $1/\sqrt{N_r}$. In the next sections, we will describe how

quantum computers can quadratically improve this dependence.

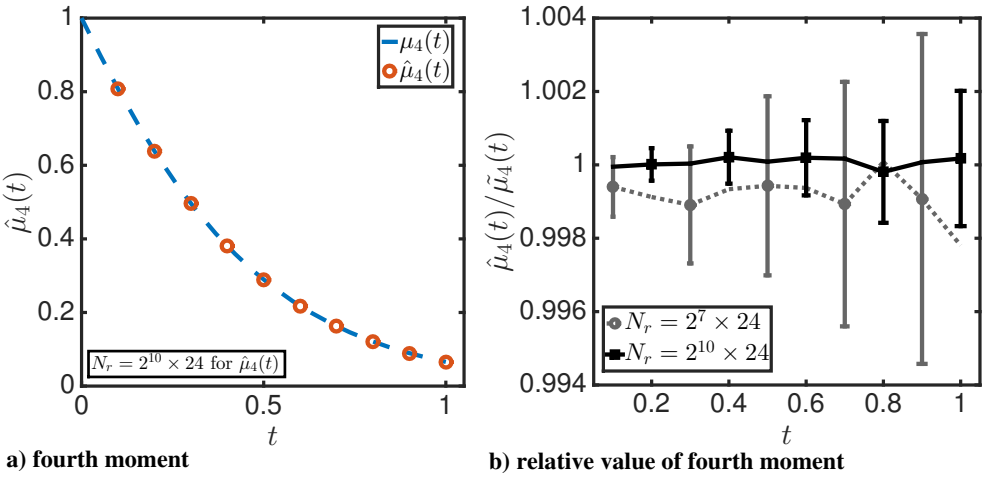


Fig. 1 MC simulations of a simple binary mixing process using Curl's model. (a) Exponential decay of the estimated 4-th central moment $\hat{\mu}_4(t)$ [Eq. (12)] as a function of time for a number of repetitions $N_r = 2^{10} \times 24$. The estimated moments are very close to the exact solution $\mu_4(t)$ (dashed line), given by Eq. (10). (b) The estimated 4-th central moment relative to a very accurate estimate $\tilde{\mu}_4(t)$ for $N_p = 10^3$. To reach a confidence level of 99.75%, the error bars include up to three estimated standard deviations of the central moment [Eq. (16)]. The standard deviation is estimated by running the MC method N_r times, for $N_r = 2^7 \times 24$ (dotted line, odd positions) and $N_r = 2^{10} \times 24$ (solid line, even positions). The relative error increases with t as both $\hat{\mu}_4(t)$ and $\tilde{\mu}_4(t)$ decay exponentially with t. The estimation error of $\hat{\mu}_4(t)$ is of order $1/\sqrt{N_r}$. Both of the figures obtained from simulations with initial PDF $P(\psi, 0) = \frac{1}{2}\delta(\psi - 1) + \frac{1}{2}\delta(\psi + 1)$. The simulation parameters are $\beta = 2$, $\omega = 1$, $\gamma = 0.5$, $t = 1$, $\Delta t = 0.1$, and $N_p = 10^3$. The initial values are set so that $\psi^k(i, 0) = -1$ for all $i \leq N_p/2$ and $\psi^k(i, 0) = +1$, otherwise.

IV. Quantum Computing and Quantum Metrology

Before we introduce our quantum algorithm, we provide a brief overview of the necessary background in quantum computing and quantum metrology. We then refer to Refs. [53, 54] and the references within this section for more details. As a note, we use the standard bra-ket notation in which a state $|\phi\rangle$ can be associated with a column vector ϕ in the complex and finite dimensional Hilbert space $\mathbb{C}^N$, and $\langle\phi|$ can be associated with $\phi^\dagger$, the conjugate transpose of ϕ [53–55].

A. Quantum states and transformations

In the circuit model of quantum computation, the fundamental unit is the qubit. A qubit's state can be in any linear superposition of $|0\rangle$ and $|1\rangle$, i.e. $|\Psi\rangle = a_0 |0\rangle + a_1 |1\rangle$, where the complex numbers a_0 and a_1 are normalized to unity: $|a_0|^2 + |a_1|^2 = 1$. The Hilbert space is $\mathbb{C}^2$. In this representation, the states in the computational basis are

$$|0\rangle \doteq \begin{pmatrix} 1 \\ \\ 0 \end{pmatrix}, \; |1\rangle \doteq \begin{pmatrix} 0 \\ \\ 1 \end{pmatrix} . \tag{17}$$

We also define the single-qubit states $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ and $|-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$. Assigned to each qubit are the Pauli (unitary) matrices

$$\sigma_0 = \mathbb{1}_2 = \begin{pmatrix} 1 & 0 \\ \\ 0 & 1 \end{pmatrix}, \sigma_x = \begin{pmatrix} 0 & 1 \\ \\ 1 & 0 \end{pmatrix}, \sigma_y = \begin{pmatrix} 0 & -i \\ \\ i & 0 \end{pmatrix}, \sigma_z = \begin{pmatrix} 1 & 0 \\ \\ 0 & -1 \end{pmatrix} . \tag{18}$$

In general, $\mathbb{1}_D$ will refer to the identity matrix of dimension D and is associated with a trivial operation. Operations on a single qubit are implemented by sequences of unitary transformations such as $R(\nu, \theta) = e^{-i\theta\sigma_\nu/2}$, and $\nu = x, y, z$. Up to a phase factor, these can be interpreted as rotations around the ν axis (rotations in the Bloch's sphere as in Fig. 2). Another useful and standard single-qubit operation used in quantum computing is the so called Hadamard transformation H, which transforms as $H|0\rangle = |+\rangle$ and $H|1\rangle = |-\rangle$, so that

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ \\ 1 & -1 \end{pmatrix} . \tag{19}$$

The state of n qubits can be represented as

$$|\Psi\rangle = \sum_{l=0}^{N-1} a_l |l\rangle = a_0 |0 \ldots 00\rangle + a_1 |0 \ldots 01\rangle + \ldots + a_{N-1} |1 \ldots 11\rangle . \tag{20}$$

The Hilbert space is $\mathbb{C}^N$ and the dimension is $N = 2^n$. $|l\rangle$ represents a state in the computational basis, where l is the corresponding binary representation. The normalization condition implies $\sum_{l=0}^{N-1} |a_l|^2 = 1$. These states can then be represented as a vector of unit length. In some cases, it will be useful to label the state each qubit independently as, for example, $|00 \ldots\rangle = |0\rangle_1 |0\rangle_2 \ldots |0\rangle_n$. The algebra associated with n-qubit systems is generated by tensor products of Pauli matrices, that

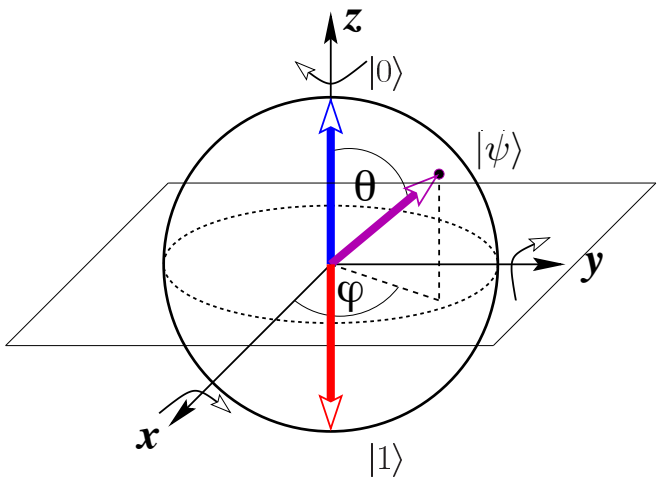


Fig. 2 A Bloch sphere representation of single-qubit unitary transformations. Up to a global phase factor, single qubit states can be represented as $|\Psi\rangle = \cos(\theta/2)\,|0\rangle + e^{i\varphi}\sin(\theta/2)\,|1\rangle$. The curved arrows indicate rotations with respect to the corresponding axis ν.

is,

$$\sigma_\nu^j = \sigma_0 \otimes \cdots \otimes \underbrace{\sigma_\nu}_{j\text{th position}} \otimes \cdots \otimes \sigma_0 \ . \tag{21}$$

These are the Pauli operators "acting" on the j-th qubit. Here, $\nu = 0, x, y, z$ and $j = 1, \ldots, n$. Note that $\sigma_0^j = \mathbb{1}_N$ for all j and is associated with a trivial operation. In quantum computing, many-qubit operations are implemented by general unitary transformations. One is typically concerned in applying such transformations (or approximations thereof) using sequences of gates drawn from a universal gate set. One commonly considered gate set is that of transformations acting on one and two qubits, such as

$$R_j(\nu,\theta) = e^{-i\theta\sigma_\nu^j/2} \ , \ R_{j,k}(\omega) = e^{-i\omega\sigma_z^j\sigma_z^k} \ . \tag{22}$$

Other universal sets of quantum gates can be obtained from the $R_j(\nu,\theta)$ and controlled operations

such as CNOT, whose representation in a basis for the four-dimensional space of two qubits is

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} . \tag{23}$$

This corresponds to a two-qubit unitary operation that "flips" the state of a target qubit depending on the state of a control qubit; that is, it performs the transformation $|00\rangle \to |00\rangle$, $|01\rangle \to |01\rangle$, $|10\rangle \to |11\rangle$, and $|11\rangle \to |10\rangle$.

For any n-qubit unitary U, we can define another $n+1$-qubit unitary transformation ${}^{\mathrm{c}}U$, which is controlled on the state of an ancillary qubit **a** being in $|1\rangle$ and transforms as follows:

$${}^{\mathrm{c}}U\,|0\rangle_{\mathrm{a}}\,|\Psi\rangle = |0\rangle_{\mathrm{a}}\,|\Psi\rangle \ , \ {}^{\mathrm{c}}U\,|1\rangle_{\mathrm{a}}\,|\Psi\rangle = |1\rangle_{\mathrm{a}}\,U\,|\Psi\rangle \ . \tag{24}$$

The CNOT transformation described above is one example of this.

In quantum mechanics, all measurable quantities have an associated Hermitian operator (the observable). In our case, we are only concerned with simple measurements of qubits in the basis $|0\rangle$ and $|1\rangle$ (i.e., the computational basis), where the measurement operators are the σ_z^j. If the quantum state is described as in Eq. (20), the probability of obtaining outcome l and projecting the state into $|l\rangle$, after a simple measurement of all qubits, is $|a_l|^2$ [56].

B. Quantum algorithms and quantum circuits

In the circuit model of quantum computing, a general quantum algorithm has three basic steps. The first step involves an initial state preparation, such as the preparation of the simple state $|0\rangle = |0\ldots0\rangle$. The second step consists of a sequence of instructions, each associated with the implementation of a gate from a universal gate set to approximate a desired n-qubit unitary operation. The final step is a projective measurement to obtain classical information that could be processed to solve a problem. The complexity of a quantum algorithm is given by the number of simple operations needed for each of the three steps. Typically, this complexity is dominated by the number of elementary unitary gates needed to prepare the initial state and the second step, since the complexity of simple measurements is assumed to be, at most, linear in n.

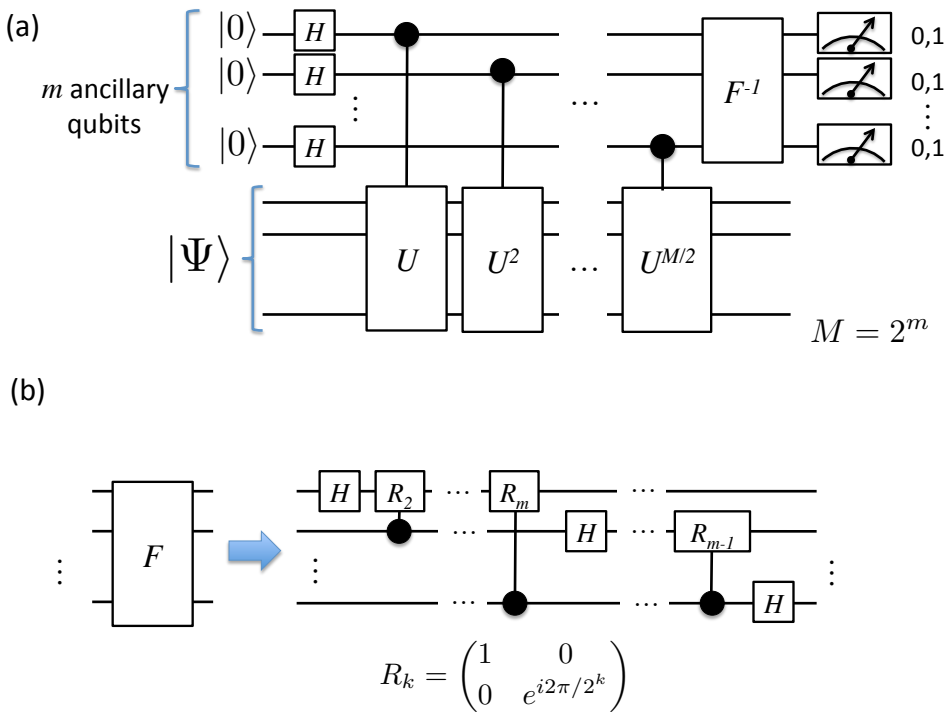


Fig. 3 (a) Quantum circuit for phase estimation (PEA). The black circles denote a controlled operation (e.g., a controlled U^k) on the corresponding state $|1\rangle$ of an ancilla qubit. The measurement outcome provides an estimate of an eigenphase or eigenvalue of U in binary representation (Sec. IV D). $F^{-1} = F^\dagger$ is the unitary transformation that corresponds to the inverse of the discrete Fourier transform (i.e., the inverse of the quantum Fourier transform). (b) Quantum circuit for the Fourier transform in terms of one and two-qubit (controlled) elementary gates. (For simplicity, the quantum circuit for F does not show a trivial swap operation that permutes the order of the qubits at the end.)

Quantum algorithms are commonly represented by quantum circuits, which are sequences of elementary unitary gates applied to an arbitrary initial state (time goes from left to right). An example of a quantum circuit is given in Fig. 3, which describes the so-called quantum phase estimation algorithm (PEA). In this case, the PEA uses the quantum Fourier transform F, for which the quantum circuit is also given in Fig. 3 (b). The PEA outputs an estimate of an eigenphase or eigenvalue of a unitary U [57]. See Sec. IV D for more details.

C. Classical and quantum computing

The class of problems that can be solved efficiently or in polynomial time with a quantum computer is referred to as BQP. A well known result states that BPP $\subseteq$ BQP, where BPP is the class of problems that can be solved in polynomial time with a classical probabilistic computer. To this end, we note that any Boolean function $F : \{0,1\}^n \to \{0,1\}^m$ can be computed with a permutation $F' : \{0,1\}^{n+m} \to \{0,1\}^{n+m}$ such that $F'(x,y) = (x, y \oplus F(x))$, where $\oplus$ is addition modulo 2. F' is then a permutation and a reversible function, and thus it can be simulated with unitary gates and a quantum circuit [58]. In fact, any permutation can be realized with sequences of permutations on three bits, and such transformations only require negation and so-called Toffoli gates. Negation is a one-qubit transformation and its matrix representation is simply given by σ_x. A Toffoli transformation is a three-qubit operation and its matrix representation is

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} . \tag{25}$$

To simulate a classical probabilistic algorithm efficiently on a quantum computer, each perfectly classical random bit can be simulated by introducing a new ancilla qubit in the state $|+\rangle_{\mathrm{a}}$. (A random bit that has probability p of being in 0 can be simulated with an ancilla qubit in the state $\sqrt{p}\,|0\rangle_{\mathrm{a}} + \sqrt{1-p}\,|1\rangle_{\mathrm{a}}$.) We can then operate controlled on the state of the ancilla and disregard it at the end of the computation. In more detail, assume that the state of a classical computer is $\sigma \in \{0,1\}^n$. We then introduce a random bit and depending on the value of the random bit we transform the state to $\sigma_0 \in \{0,1\}^n$ or $\sigma_1 \in \{0,1\}^n$ using reversible operations, as described before. These states can be obtained via the action of a permutation. To simulate this transformation on

a quantum computer, we can implement a unitary transformation that operates as follows:

$$|+\rangle_{\rm a}\,|\sigma\rangle \to \frac{1}{\sqrt{2}}\left(|0\rangle_{\rm a}\,|\sigma_0\rangle + |1\rangle_{\rm a}\,|\sigma_1\rangle\right) \ . \tag{26}$$

It follows that a simple measurement of the ancilla qubit in Eq. (26) provides the outcome 0 or 1 with probability exactly $1/2$, thereby simulating the classical probabilistic process. In either case, the state of the quantum computer is projected into $|\sigma_0\rangle$ or $|\sigma_1\rangle$, respectively.

D. Quantum computing and metrology

An important application of quantum processing of information is precision sensing or quantum-enhanced metrology (QM) (c.f., [59] and references therein). The goal of QM is to obtain properties of quantum states as precisely as possible given the available resources. Consider, for example, the problem of obtaining the probability p by making measurements on the single qubit state $|\Psi\rangle = \sqrt{p}\,|0\rangle + \sqrt{1-p}\,|1\rangle$. One possible way to estimate p is by repeated state preparations and by counting the frequency of outcome 0 after measurement. This procedure is similar to that of estimating the probability that a biased coin lands in tails (or heads) by repeated coin flips. Chebyshev's inequality states that the uncertainty in the estimation of p scales as $1/\sqrt{N_r}$, where N_r is the number of repetitions. Quantum computers, however, can achieve the same precision in the estimation of p using the unitary that prepares $|\Psi\rangle$ only order of $\sqrt{N_r}$ or $1/p$ times [5, 6, 35]. This is the so-called QM limit.

One method to achieve the QM limit is as follows. Let U' be the single qubit unitary transformation that implements $U'\,|0\rangle = \sqrt{p}\,|0\rangle + \sqrt{1-p}\,|1\rangle$. Using the Pauli matrices, we can represent U' as $e^{-i\theta\sigma_y/2}$, where $\theta = 2\arccos(\sqrt{p})$. We also define $U = e^{-i\theta\sigma_z/2}$, which is basically U' conjugated by the unitary that transforms $\sigma_y \to \sigma_z$. The quantum algorithm will produce an estimate of θ within precision $\epsilon > 0$, which can be translated to an estimate of p with the same order of precision. The estimate of θ is $\hat{\theta}$, which we represent in binary as

$$\begin{aligned}\hat{\theta} &= 2\pi[.b'_1, b'_2, \ldots b'_m] \\ &= \pi(b'_1 + b'_2/2 + \ldots b'_m/2^{m-1}) \ .\end{aligned} \tag{27}$$

Here, $b'_i \in \{0,1\}$ specifies the bits of the number in the binary representation, and we choose $m = O(\log_2(1/\epsilon))$ so that the desired ϵ precision is achieved. The quantum algorithm is defined in

m basic steps, where each step j results in the outcome b'_{m-j+1} (i.e., we start by estimating the least significant bit and move towards the most significant ones). The single-qubit PEA is

Input: A single-qubit unitary $U = e^{-i\theta\sigma_z/2}$ and a precision parameter $\epsilon > 0$.

1. Obtain the smallest integer m such that $M \geq 2\pi/\epsilon$, with $M = 2^m$.

2.

2.1 Prepare the single-qubit state $|+\rangle$ and apply U, $M/2$ times.

2.2 Apply a Hadamard transformation and measure the qubit in the computational basis.

Let $b'_m \in \{0, 1\}$ be the measurement outcome.

3. Do the following for each $k = (m-1), \ldots, 1$:

3.1 Prepare the single qubit state $|+\rangle$ and apply U, 2^{k-1} times.

3.2 Compensate the phase of $|1\rangle$ by $e^{-i\pi[.b'_{k+1}\ldots b'_m]}$.

3.3 Apply a Hadamard transformation and measure the qubit in the computational basis.

Let $b'_k \in \{0, 1\}$ be the measurement outcome.

Output: An estimate of θ as $\hat{\theta} = 2\pi[.b'_1 \ldots b'_m]$.

The probability that this quantum algorithm returns an m-bit estimate $\hat{\theta}$ is, in general,

$$\Pr(\hat{\theta}) = \frac{1}{4^m} \left| \frac{e^{i2^m\theta} - 1}{e^{i(\theta-\hat{\theta})} - 1} \right|^2 . \tag{28}$$

In particular, if θ can be exactly represented by $m-1$ bits as in Eq. (27) (i.e., θ is an m-th root of unity), the quantum algorithm provides an estimate that is exact: $\Pr(\hat{\theta} = \theta) = 1$ and $2^m\theta = 0 \mod (2\pi)$ in that case. In general, this quantum algorithm returns one of the two best m-bit approximations of θ with probability (confidence level) at least 0.81 [60]. This algorithm is a version of the PEA of Fig. 3 where the (inverse) quantum Fourier transform is implemented sequentially [61] attaining the same output.

In general, the choice of m ensures that these approximations are within precision ϵ. For constant confidence level and precision $\epsilon = O(1/M)$, the quantum algorithm requires $M = 2^m$ uses of U.

This is a quadratic cost improvement over standard methods [62]. It is also possible to arbitrarily increase the confidence level of the estimation to $c < 1$ by repetition as follows. Let $\hat{\theta}_1, \ldots, \hat{\theta}_L$ be L estimates of the phase θ obtained by L independent executions of the PEA. Let θ_l and θ_r be the two closest m-bit approximations of θ. Then, the probability that $\hat{\theta}_i \notin [\theta_l, \theta_r]$ is bounded from above by $p_f = 0.19$. We will then obtain the estimate $\hat{\theta}$ as the median of the L estimates. By doing so, the probability that $\hat{\theta} \notin [\theta_l, \theta_r]$ can be bounded by [6, 63]

$$\frac{1}{2}\left(2\sqrt{p_f(1-p_f)}\right)^L \leq \frac{1}{2}(0.8)^L \ . \tag{29}$$

Then, $L = O(|\log(1-c)|)$ repetitions suffice for a confidence level c. The number of times the unitary U is used is

$$N_r = L \times M \ . \tag{30}$$

The previous single-qubit PEA can be simply generalized to provide the eigenphase of a unitary U acting on n-qubit states. Let $|\Psi\rangle = V|00\ldots0\rangle$ be the eigenvector of U that satisfies $U|\Psi\rangle = e^{i\theta}|\Psi\rangle$. We also define the unitary ${}^{\mathrm{c}}U$, which implements U controlled on the state of an ancilla qubit being in $|1\rangle_{\mathrm{a}}$ or does nothing if the state is $|0\rangle_{\mathrm{a}}$. The PEA to estimate θ is:

Input: n-qubit unitaries U and V, and a precision parameter $\epsilon > 0$.

1. Obtain the smallest integer m such that $M \geq 2\pi/\epsilon$, with $M = 2^m$.

2.

2.1 Prepare $|\Psi\rangle$ and the single-qubit ancilla state $|+\rangle_{\rm a}$, and apply ${}^{\rm c}U$, $M/2$ times.

2.2 Apply a Hadamard transformation and measure the ancilla qubit in the computational basis.

Let $b'_m \in \{0, 1\}$ be the measurement outcome.

3. Do the following for each $k = (m-1), \ldots, 1$:

3.1 Prepare the single-qubit ancilla state $|+\rangle_{\rm a}$ and apply ${}^{\rm c}U$, 2^{k-1} times.

3.2 Compensate the phase of $|1\rangle_{\rm a}$ by $e^{-i\pi[.b'_{k+1}\ldots b'_m]}$.

3.3 Apply a Hadamard transformation and measure the ancilla qubit in the computational basis.

Let $b'_k \in \{0, 1\}$ be the measurement outcome.

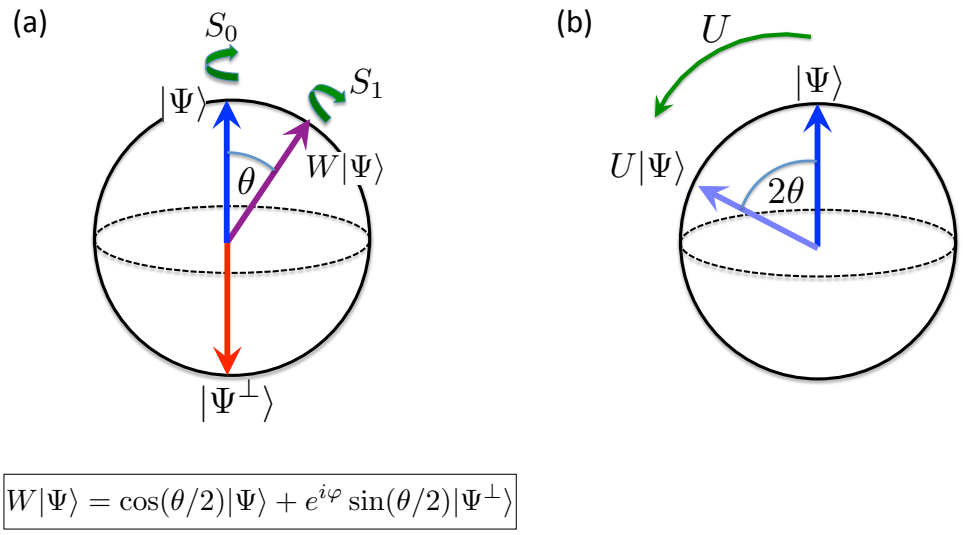
Output: An estimate of θ as $\hat{\theta} = 2\pi[.b'_1 \ldots b'_m]$.

This algorithm implements the same transformation and provides the same output as that of the PEA in Fig. 3 [6]. As in the single-qubit case, it can be shown that the probability that the algorithm outputs $\hat{\theta}$ is given by Eq. (28). Thus, one of the two closest m-bit approximations of θ is obtained with probability of, at least, 0.81. For arbitrary confidence level, the algorithm needs to be repeated $L = O(|\log(1-c)|)$ times. The complexity of the algorithm is then mainly dominated by the number of uses of ${}^{\rm c}U$, which is N_r, and L uses of V. The gate complexity of the algorithm is obtained after decomposing ${}^{\rm c}U$ and V as a sequence of elementary one and two-qubit gates.

The previous algorithm can also be used when the input state $|\Psi\rangle$ is not an eigenstate of U but rather a linear combination of eigenstates; that is

$$|\Psi\rangle = \sum_j c_j |\Psi_j\rangle \ , \tag{31}$$

where $c_j \in \mathbb{C}$ and $|\Psi_j\rangle$ satisfies $U|\Psi_j\rangle = e^{i\theta_j}|\Psi_j\rangle$. Note that $\sum_j |c_j|^2 = 1$. The output of the algorithm is then an estimate $\hat{\theta}_j$ of θ_j with probability given by $|c_j|^2$, which is the population of $|\Psi\rangle$ in the corresponding eigenstate.



$$\boxed{W|\Psi\rangle = \cos(\theta/2)|\Psi\rangle + e^{i\varphi}\sin(\theta/2)|\Psi^\perp\rangle}$$

Fig. 4 Bloch's sphere representation of the two-dimensional vector space spanned by $|\Psi\rangle$ and $W|\Psi\rangle$. Under the assumption that $w = \langle\Psi|W|\Psi\rangle \geq 0$, we obtain $w = \cos(\theta/2)$. (a) Representation of the two reflections S_0 and $S_1 = WS_0W^\dagger$, with S_0 defined in Eq. (32). (b) Representation of the effective rotation $U = S_0S_1$ for an angle of 2θ. The eigenvalues of U are $e^{\pm i\theta}$ and its eigenphases are $\pm\theta$.

The PEA can also be used as a subroutine to obtain expectation values of operators in n-qubit states with minimal prior knowledge [6, 9]. To illustrate this, let W be an n-qubit unitary operation and $w = \langle\Psi|W|\Psi\rangle$ be the expectation value of W in the pure n-qubit state $|\Psi\rangle = V|00\ldots0\rangle$. For simplicity, assume that $w \geq 0$. (The analysis for the general case where $w \in \mathbb{C}$ is slightly more involved and can be found in [6].) The quantum states $|\Psi\rangle$ and $W|\Psi\rangle$ span a vector (Hilbert) space of dimension 2, as in Fig. 4. In this case, $W|\Psi\rangle = \cos(\theta/2)|\Psi\rangle + e^{i\varphi}\sin(\theta/2)|\Psi^\perp\rangle$, where $|\Psi^\perp\rangle$ is the state orthogonal to $|\Psi\rangle$ in the subspace. Thus, $w = \cos(\theta/2)$, with $\theta \leq \pi$. The "trick" to obtain w is to design a unitary operation that has θ as eigenphase and then use the PEA.

We first consider the unitary operation S_0, which implements a reflection over the state $|\Psi\rangle$; that is

$$S_0|\Psi\rangle = -|\Psi\rangle \ , \ S_0|\Psi^\perp\rangle = |\Psi^\perp\rangle \ . \tag{32}$$

Equivalently, we can write $S_0 = \mathbb{1}_{2^n} - 2|\Psi\rangle\langle\Psi| = \mathbb{1}_{2^n} - 2V|00\ldots0\rangle\langle00\ldots0|V^\dagger$. The implication is that S_0 can be implemented by first applying $V^\dagger$, then applying a reflection over the simple n-qubit

state $|00\ldots0\rangle$, and then applying V. The reflection over $|00\ldots0\rangle$ can be performed using standard techniques with a number of one and two-qubit elementary gates that is linear in n [53]. Then, the gate complexity of S_0 is twice the gate complexity of V and additionally $O(n)$ gates.

Next we consider the unitary operation S_1, which implements a reflection over $W|\Psi\rangle$. This is simply $S_1 = WS_0W^\dagger$, and the gate complexity of S_1 is that of S_0 plus twice the gate complexity of W. The composition of the two reflections, $U = S_0S_1$, is then a rotation in the two-dimensional Hilbert space by an angle of 2θ. Thus, its eigenvalues in that subspace are $e^{\pm i\theta}$, and the PEA can be used to estimate θ and thus w. Additionally, it can be shown that

$$|\Psi\rangle = \frac{1}{\sqrt{2}}\left(|\Psi_+\rangle + |\Psi_-\rangle\right) , \tag{33}$$

where $|\Psi_\pm\rangle$ are such that $U|\Psi_\pm\rangle = e^{\pm i\theta}|\Psi_\pm\rangle$, i.e., the eigenstates of U. It implies that, if the initial state is $|\Psi\rangle$ and we execute the PEA, we will obtain an estimate of θ or $-\theta$ with probability $1/2$, respectively. Since we are interested in $w = \cos(\theta/2)$, any of these estimations suffices. The quantum algorithm to estimate w is:

Input: n-qubit unitaries W and V, and a precision parameter $\epsilon > 0$.

1. Obtain the smallest integer m such that $M \geq 2\pi/\epsilon$, with $M = 2^m$.

2.

2.1 Prepare $|\Psi\rangle$ and the single-qubit ancilla state $|+\rangle_{\rm a}$, and apply ${}^{\rm c}U$, $M/2$ times. Here, $U = S_0 S_1$.

2.2 Apply a Hadamard transformation and measure the ancilla qubit in the computational basis. Let $b'_m \in \{0, 1\}$ be the measurement outcome.

3. Do the following for each $k = (m-1), \ldots, 1$:

3.1 Prepare the single-qubit ancilla state $|+\rangle_{\rm a}$ and apply ${}^{\rm c}U$, 2^{k-1} times.

3.2 Compensate the phase of $|1\rangle_{\rm a}$ by $e^{-i\pi[.b'_{k+1}\ldots b'_m]}$.

3.3 Apply a Hadamard transformation and measure the ancilla qubit in the computational basis. Let $b'_k \in \{0, 1\}$ be the measurement outcome.

Output: $\hat{w}$, an estimate of $w = \langle\Psi| W |\Psi\rangle$ as $\cos(\hat{\theta}/2)$, with $\hat{\theta} = 2\pi[.b'_1 \ldots b'_m]$.

Since $\hat{\theta}$ is an estimate of either θ or $-\theta$ within precision ϵ, the precision ϵ' in the estimation of w at first order in ϵ is $O((\epsilon/2)\sin(\hat{\theta}/2))$. It follows that better precision estimates are obtained when w is near 1 (i.e, θ and $\hat{\theta}$ are near zero). To bound the error in these cases, we can use the inequality

$$|\cos(\hat{\theta}/2) - \cos(\theta/2)| \leq \epsilon' = |\cos((\hat{\theta} + \epsilon)/2) - \cos(\hat{\theta}/2)| \,, \tag{34}$$

which is valid when $0 \leq \hat{\theta} \leq \pi$ and $0 < \epsilon \leq 1$. A similar bound can be obtained for $-\pi \leq \hat{\theta} \leq 0$.

That is, the above quantum algorithm produces an estimate $\hat{w}$ that satisfies

$$\Pr(|\hat{w} - w| \leq \epsilon') \geq 0.81 \,. \tag{35}$$

As before, we can arbitrarily increase the confidence bounds to $c < 1$ in the estimation by obtaining $L = O(|\log(1-c)|)$ independent estimates of θ and computing the median– see Eq. (29).

Last, we focus on the estimation of expectation values $a = \langle\Psi| A |\Psi\rangle$, where A is an observable (i.e., $A = A^\dagger$) but not necessarily a unitary operation. There are several ways to use the previous

algorithm to obtain a depending on A; see [6] for an example. If $A \geq 0$ and $\|A\| \leq 1$, in some cases it is possible to construct a unitary that acts as

$$W |\Psi\rangle |0\rangle_{\rm a} = A |\Psi\rangle |0\rangle_{\rm a} + B |\Psi\rangle |1\rangle_{\rm a} \ , \tag{36}$$

where a is an ancillary qubit. Examples of such unitaries W have been considered in recent quantum algorithms for various problems [64, 65]. Then,

$$\begin{aligned} a &= \langle\Psi| A |\Psi\rangle \\ &= \langle\Psi| \langle 0|_{\rm a} W |\Psi\rangle |0\rangle_{\rm a} \ , \end{aligned} \tag{37}$$

and the problem reduces to the estimation of the expectation value of the unitary W in the state $|\Psi\rangle |0\rangle_{\rm a}$. See [6, 9, 64, 65] for addressing a more general case.

V. A quantum algorithm for the C/D model

We now consider a quantum algorithm to simulate the same problem as the classical MC method described in Sec. III. We will show that for accurate estimation of particular properties of ψ at a given time t, this algorithm provides a quadratic speedup over the corresponding classical method. We will introduce the algorithm in subsection V A and follow this with the simulation of the corresponding PEA for binary mixing in subsection V C. While classical MC methods can be parallelized by running different repetitions at the same time, the quantum algorithm presented in subsection V A is sequential. Nevertheless, we explain a potential way to deal with parallelization in subsection V D.

A. General statement of the algorithm

To build our quantum algorithm, we first focus on the preparation of the initial quantum state $|\Psi\rangle$. The amplitudes of this quantum state encodes all the information obtained by the MC algorithm of Sec. III. It is prepared by a sequence of elementary gates that represent reversible operations that simulate the random processes in MC, as described in Sec. IV C. That is,

$$\begin{aligned} |\Psi\rangle &= V |00\ldots0\rangle \\ &= \sum_{\psi_1,\ldots,\psi_{N_p}} \sqrt{Q(\psi_1,\ldots,\psi_{N_p})} \, |\psi_1,\ldots,\psi_{N_p}\rangle \, |\xi_{\psi_1,\ldots,\psi_{N_p}}\rangle \ . \end{aligned} \tag{38}$$

The probabilities $Q(\psi_1, \ldots, \psi_{N_p}, t_j)$ are exactly those of the MC algorithm at step j, i.e., they are the probabilities that $\psi^k(i, t_j) = \psi_i$:

$$Q(\psi_1, \ldots, \psi_{N_p}) = Q(\psi^k(1, t_j) = \psi_1, \ldots, \psi^k(N_p, t_j) = \psi_{N_p}) \tag{39}$$

Each $|\psi_1, \ldots, \psi_{N_p}\rangle$ is a state in the computational basis having ψ_i in binary representation and $|\xi_{\psi_1,\ldots,\psi_{N_p}}\rangle$ is a quantum state that contains information about all intermediate calculations and will be discarded. The algorithm for initial state preparation is:

Input: t, β, ω, Δt, N_p, N_r

1. Obtain $N_t = \lceil t/\Delta t \rceil$, $N_s = \lceil \beta\omega\Delta t N_p \rceil$.

2. Obtain a description of all simple classical gates $v_1, \ldots, v_T$ involved in the MC algorithm of Sec. III.

3. Obtain the one and two-qubit (reversible) gates $\tilde{v}_1, \ldots, \tilde{v}_T$ that are reversible versions of the v_i (Sec. IV C).

4. Construct and implement a unitary $V = \tilde{v}_T \ldots \tilde{v}_1$ on the initial state $|00\ldots0\rangle$.

Output: The quantum state $|\Psi\rangle = V|00\ldots0\rangle$.

A measurement on $|\Psi\rangle$ of the register that encodes the ψ_i will output $\psi_1, \ldots, \psi_{N_p}$ with probability $Q(\psi_1, \ldots, \psi_{N_p})$, as expected. However, such measurements will not be performed directly in this algorithm - instead we employ the quantum metrology techniques of Sec. IV D to obtain better estimates. The number of qubits n needed to represent $|\Psi\rangle$ scales with the number of bits needed to implement the classical MC method. Also, the complexity of preparing $|\Psi\rangle$, i.e., the number of gates to implement V, is similar to that of a single run of the MC algorithm since each simple operation in MC is replaced by an equivalent reversible operation in the quantum algorithm. Our objective is to reduce the resource requirements in terms of N_r, the number of repetitions of the MC method.

Our goal is to estimate properties of the distribution $Q(\psi^k(1, t_j), \ldots, \psi^k(N_p, t_j))$. Assume, for example, that we aim at obtaining the l-th central moment of this distribution as defined by Eq. (6).

In MC, the estimation of the l-th central moment is obtained via Eq. (12), which in the limit of $N_r \to \infty$, they become

$$\frac{1}{N_p} \sum_{\psi_i,\ldots,\psi_{N_p}} Q(\psi_1,\ldots,\psi_{N_p}) \left[(\psi_1 - \tilde{E}[\psi^k(t_j)])^l + \ldots + (\psi_{N_p} - \tilde{E}[\psi^k(t_j)])^l \right] . \tag{40}$$

Here,

$$\tilde{E}[\psi^k(t_j)] := \frac{1}{N_p} \sum_{\psi_1,\ldots,\psi_{N_p}} Q(\psi_1,\ldots,\psi_{N_p})(\psi_1 + \ldots + \psi_{N_p}) . \tag{41}$$

(In the case of binary mixing, we can assume $\tilde{E}[\psi^k(t_j)] = 0$.) It is then simple to construct a (diagonal) observable A such that

$$\langle\Psi| A |\Psi\rangle = \frac{1}{N_p} \sum_{\psi_1,\ldots,\psi_{N_p}} Q(\psi_1,\ldots,\psi_{N_p})((\psi_1)^l + \ldots + (\psi_{N_p})^l) . \tag{42}$$

The observable has the property

$$A|\psi_1,\ldots,\psi_{N_p}\rangle = \frac{1}{N_p}((\psi_1)^l + \ldots + (\psi_{N_p})^l)|\psi_1,\ldots,\psi_{N_p}\rangle . \tag{43}$$

Under the assumption $|\psi_i| \leq 1$, as is the case of binary mixing, it is simple to show the existence of a unitary W that implements [Eq. (37)]

$$W \left|\psi_1,\ldots,\psi_{N_p}\right\rangle |0\rangle_{\rm a} = A \left|\psi_1,\ldots,\psi_{N_p}\right\rangle |0\rangle_{\rm a} + |\phi^\perp\rangle |1\rangle_{\rm a} , \tag{44}$$

where $|\phi^\perp\rangle$ is an irrelevant quantum state. For example, in block-matrix form,

$$W = \begin{pmatrix} A & \sqrt{1-A^2} \\ \sqrt{1-A^2} & -A \end{pmatrix} , \tag{45}$$

where the first (second) block in the diagonal corresponds to the subspace where the ancillary state is $|0\rangle_{\rm a}$ ($|1\rangle_{\rm a}$). Keeping all other variables constant (including the number of qubits needed to represent each ψ_i), the gate complexity of W is polynomial in N_p. This gate complexity may be negligible when compared to the gate complexity of V. The techniques invoked to implement W are standard in quantum computing [53]. Equation (44) implies

$$\begin{aligned} w &= \langle\Psi| \langle 0|_{\rm a} W |\Psi\rangle |0\rangle_{\rm a} \\ &= \frac{1}{N_p} \sum_{\psi_1,\ldots,\psi_{N_p}} Q(\psi_1,\ldots,\psi_{N_p})((\psi_1)^l + \ldots + (\psi_{N_p})^l) . \end{aligned} \tag{46}$$

It is then simple to reduce the problem of estimating the l-th central moment [Eq. (40)] to that of estimating the expectation value of a unitary W [Eq. (46)].

For some mixing problems, such as binary mixing, the l-th central moment may decay rapidly as a function of t. This translates to a small value of w and thus the precision of the estimation is of order $\epsilon/2$ (Sec. IV D). To improve the precision, we can use a simple trick to shift the estimate of the l-th central moment by computing the expected value of a unitary W that is close to 1 [see Eq. (34)]. In this case, instead of using W as in Eq. (44), we can define W as

$$W|\psi_1,\ldots,\psi_{N_p}\rangle\,|0\rangle_{\rm a} = (1-A)|\psi_1,\ldots,\psi_{N_p}\rangle\,|0\rangle_{\rm a} + |\eta^{\perp}\rangle\,|1\rangle_{\rm a}\ , \tag{47}$$

where $|\eta^{\perp}\rangle$ is also an irrelevant quantum state. The estimation of $w = \langle\Psi|\,\langle 0|_{\rm a}\,W\,|\Psi\rangle\,|0\rangle_{\rm a}$ within precision ϵ_Q results in the estimation of the l-th central moment within the same order of precision.

We are ready to use the techniques of quantum metrology to obtain the l-th central moment. Our main result is the following quantum algorithm:

Input: l, t, β, ω, Δt, N_p, ϵ

1. Obtain $N = \lceil t/\Delta t \rceil$, $N_s = \lceil \beta\omega\Delta t N_p \rceil$ and the smallest integer m such that $M \geq 2\pi/\epsilon$, with $M = 2^m$.

2. Construct the unitary V to prepare $|\Psi\rangle$ as in Eq. (38).

3. Construct the unitary W as in Eq. (44) or Eq. (47).

4. Construct the unitary $U = S_0 S_1 = S_0 W S_0 W^\dagger$, where $S_0 = \mathbb{1}_{2^n} - 2\,|\Psi\rangle\langle\Psi|$ is the reflection operator.

5.

5.1 Prepare $|\Psi\rangle$ and the single-qubit ancilla state $|+\rangle_{\rm a}$, and apply ${}^{\rm c}U$, $M/2$ times.

5.2 Apply a Hadamard transformation and measure the ancilla qubit in the computational basis. Let $b'_m \in \{0,1\}$ be the measurement outcome.

6. Do the following for each $k = (m-1), \ldots, 1$:

6.1 Prepare the single-qubit ancilla state $|+\rangle_{\rm a}$ and apply ${}^{\rm c}U$, 2^{k-1} times.

6.2 Compensate the phase of $|1\rangle_{\rm a}$ by $e^{-i\pi[.b'_{k+1}\ldots b'_m]}$.

6.3 Apply a Hadamard transformation and measure the ancilla qubit in the computational basis. Let $b'_k \in \{0,1\}$ be the measurement outcome.

Output: An estimate of the l-th central moment as $\cos(\hat{\theta}/2)$, with $\hat{\theta} = 2\pi[.b'_1 \ldots b'_m]$.

The confidence level for the estimation is bounded from below by 0.81 and can be arbitrarily increased by L repeated estimates as described in Sec. IV D [Eq. (29)].

B. Complexity

In this section we analyze the complexity of the previous algorithms. For simplicity, we disregard logarithmic factors in the order notation. The complexity to prepare the initial state $|\Psi\rangle$ is given by the number of elementary gates to implement V. As V is constructed using reversible versions of the operations used in the MC method, it is reasonable to assume that the complexity of V for

the C/D model is of order $O(N_t N_s) = O(t\beta\omega N_p)$, i.e., the number of simple operations in a single MC run. The complexity of W is determined by the complexity of computing the corresponding function of $\psi_1, \ldots, \psi_{N_p}$ (i.e., an estimate of the l-th central moment) and the number of gates to implement W is then $O((N_p)^q)$ for some $q > 0$. The complexity of W may then be significantly smaller than that of V. As U makes two calls to V and two calls to $V^\dagger$, the complexity of U is also $O(t\beta\omega N_p)$. Our quantum algorithm uses U a total of M times and its overall complexity is then $O(t\beta\omega N_p/\epsilon)$. This result has to be compared with that of Sec. III A, where the dependence on ϵ is quadratically worse. As discussed, to reach arbitrary confidence level c, the overhead is a multiplicative factor $O(|\log(1-c)|)$.

C. Example: Quantum algorithm for binary mixing

We simulate our main quantum algorithm for the binary mixing problem to compare its performance with that of the MC method in Sec. III. By simulation of a quantum algorithm we mean a classical procedure that allows us to sample from the same outcomes as those provided by a measurement performed in the quantum state of a quantum computer. Typically, such a procedure is inefficient, having a complexity that is exponential in the number of qubits, and classical computer simulations can only be performed when the number of qubits is less than or of the order of 40. This is far fewer than the number of qubits that would be required to execute our quantum algorithm. Nevertheless, here we can simulate the estimation process because of the simplicity of the problem and our accurate knowledge of the distribution of the measurement outcome in the quantum algorithm due to our efficient classical MC simulation.

We consider the same binary mixing problem of Sec. III B and use the same parameters. To reach the same confidence level $c = 99.75\%$ as MC, the quantum algorithm has to be invoked several times. Each time we obtain an estimate of the phase and then compute the median of the estimated phases. Using Eq. (29), the number of repetitions is

$$L \geq \frac{\log(2 \times (1 - 0.9975))}{\log(0.8)} , \tag{48}$$

and we can choose $L = 24$. This is the reason why we use the convenient factorization of N_r as $2^m \times 24$.

Our classical simulations are implemented as follows. There are $L = 24$ steps and each step returns a phase $2\hat{\theta}_i$ according to the same probability distribution as that if we were to run the main quantum algorithm of Sec. V. Because of the way we construct the classical sampling method, we can assume $0 \leq \theta/2 < \pi/2$, and then $0 \leq 2\theta < 2\pi$. In the quantum algorithm, this could be done by replacing $U \to U^2$ without changing the complexity of the PEA. The probability distribution associated with $2\hat{\theta}_i$ is given by Eq. (28) (replacing $\theta \to 2\theta$ and $\hat{\theta} \to 2\hat{\theta}$), which requires knowledge of the true value of θ (i.e., at infinite precision). In our case, we are interested in obtaining an estimate to the 4-th central moment, namely $\hat{\mu}_4(t)$. As $\mu_4(t)$ decays exponentially with t, we define the unitary W via Eq. (44) when $\mu_4(t) > 1/2$, and via Eq. (47) when $\mu_4(t) \leq 1/2$ ($l = 4$). This would allow us to reduce the error as explained in Sec. IV D, Eqs. (34) and (35).

We first obtain $\tilde{\mu}_4(t)$, which is a very accurate estimate of $\mu_4(t)$ by applying the MC techniques of Sec. III B and using $N_r = 2^{20} \times 60$ times. Note that this would not be possible in more general mixing problems, which is the reason why we may need the quantum algorithm to obtain much better precision. We let $\tilde{\mu}_4(t)$ be such an accurate estimate and then obtain the actual θ as

$$\theta/2 = \begin{cases} \arccos(1 - \tilde{\mu}_4(t)) & \text{if } \tilde{\mu}_4(t) \leq 1/2 \\ \arccos(\tilde{\mu}_4(t)) & \text{if } \tilde{\mu}_4(t) > 1/2 \, . \end{cases}$$

We write $2\hat{\theta} = 2\pi[.b'_1 \dots b'_m]$ for the m bit representation of the estimate of 2θ. To sample from the distribution of Eq. (28), after replacing $\theta \to 2\theta$ and $\hat{\theta} \to 2\hat{\theta}$, we proceed as follows. After simple calculations, it can be shown that

$$\Pr(b'_m = 0) = \frac{1}{2}\left(1 + \cos(M\theta)\right) \ , \ \Pr(b'_m = 1) = 1 - \Pr(b'_m = 0), \tag{49}$$

where $M = 2^m$ and $M \geq 2\pi/\epsilon$. The sampling probabilities for the remaining bits are obtained recursively as follows. For $k = m-1, \dots, 1$, we let

$$\Pr(b'_k = 0) = \frac{1}{2}\left(1 + \cos(2^k\theta - \pi[.b'_{k+1} \dots b'_m])\right) \ , \ \Pr(b'_k = 1) = 1 - \Pr(b'_k = 0) \, . \tag{50}$$

This provides a simple way to sample from the desired distribution of Eq. (28) by sampling each bit according to a distribution that depends on the outcome of previous bits.

In Fig. 5 (a), we provide the quantum-algorithm simulation results for the estimate of the 4-th central moment, $\hat{\mu}_4(t)$. We used $m = 10$ bits of precision and $L = 24$ repetitions; that is,

$N_r = 2^{10} \times 24$. As in Sec. III B, we observe that the 4-th central moment decays exponentially in time. In Fig. 5 (b), we compare $\hat{\mu}_4(t)$ with $\tilde{\mu}_4(t)$, which is very close to the exact solution when the number of particles is $N_p = 10^3$, and for different values of m. When $t \geq 0.3$, the quantum algorithm estimates the value $1 - \mu_4(t)$, rather than $\mu_4(t)$, to obtain smaller error estimates. To obtain the error bars, we note that if $2\hat{\theta}$ is an estimate of 2θ within precision $2\pi/2^m$, then Eq. (34) implies

$$\epsilon_Q = \left|\cos((\hat{\theta} + \epsilon/2)/2) - \cos(\hat{\theta}/2)\right| \tag{51}$$

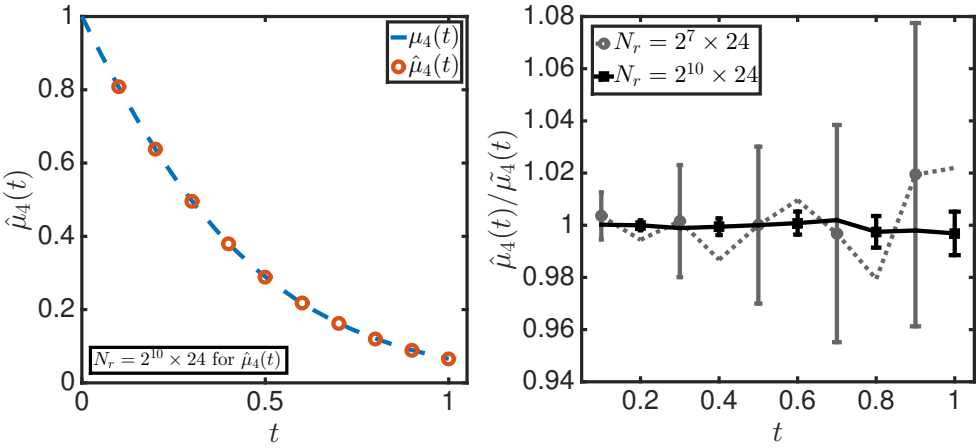
That is, we replaced ϵ by $\epsilon/2 \geq 2\pi/2^{m+1}$ in Eq. (34), since we are estimating 2θ within precision ϵ. The results shown in Fig. 5 should be compared with those in Fig. 1.

In Fig. 6, we compare the errors output by the classical MC method (ϵ_C) and our PEA (ϵ_Q). The results are for the 4-th central moment of the binary mixing model described above, for different values of t and N_r. The errors were obtained from Eqs. (15) and (51), respectively. The different scalings are clear, showing the advantages of the quantum algorithm as N_r becomes larger.

Moreover, which algorithm provides results in a shorter time will depend on the speed of the hardware, as well as prefactors associated with the specific implementation of both classical and quantum algorithms. However, the power of quantum computing is clearly demonstrated in the very different scaling of the precision with the number of repetitions. It is this change in scaling that represents what is usually referred to as the quantum speed-up, and which gives significant advantages for high-precision parameter estimation.

D. Parallelization

While classical MC methods have a poor complexity dependence on the precision parameter ϵ, one important feature is that they can be parallelized somewhat easily. Here, we investigate the extent to which our quantum algorithms that are based on phase estimation can be parallelized. To this end, we follow and adapt the results in [6] to the problem of turbulent mixing. At its core, the advantage of our quantum enhanced methods is due to two facts: i) the possibility to reduce the problem to the estimation of the phase θ of a unitary operator and ii) the possibility to encode information about $M\theta$ on a quantum state, $M = O(1/\epsilon)$, using resources that are almost linear in



a) fourth moment **b) relative value of fourth moment**

Fig. 5 Quantum-algorithm simulation of a simple binary mixing process using Curl's model with the same simulation parameters in Fig. 1. (a) Exponential decay of the estimated 4-th central moment $\hat{\mu}_4(t)$, as a function of time for a number of state preparations $N_r = 2^{10} \times 24$ [Eq. (30)]. This was obtained as $\hat{\mu}_4(t) = \cos(\hat{\theta}/2)$ $(t < 0.3)$ or $\hat{\mu}_4(t) = 1 - \cos(\hat{\theta}/2)$ $(t \geq 0.3)$, where $\hat{\theta}$ is the phase estimate obtained by the quantum PEA. The estimated moments are very close to the exact solution $\mu_4(t)$ (dashed line), given by Eq. (10). (b) The estimated 4-th central moment relative to a very accurate estimate $\tilde{\mu}_4(t)$ for $N_p = 10^3$. The data shown here are for $N_r = 2^7 \times 24$ (dotted line, odd positions) and $N_r = 2^{10} \times 24$ (solid line, even positions). To reach a confidence level of 99.75 %, the error bars were obtained as Eq.(51). The relative error increases with t as both $\hat{\mu}_4(t)$ and $\tilde{\mu}_4(t)$ decay exponentially with t. The estimation error of $\hat{\mu}_4(t)$ is of order $1/N_r$. Note that in (b) we use a different scale to that shown in Fig.1(b), and that it is not meaningful to compare the quantum and classical algorithms based on these figures alone, as the algorithms would run on different hardware. The advantage of the quantum algorithm is in the scaling with N_r, which we plot in Fig. 6, and discuss in detail in the text.

M. Under reasonable assumptions, obtaining an estimate of a function $f(M\theta)$ within precision Δf usually results in an estimate of θ within precision of order $\Delta f/M$.

As constructed, the state $|\Psi\rangle$ of Eq. (38) is an equal linear combination of eigenstates of $U = S_0 S_1$ with eigenvalues $e^{\pm i\theta}$ [Eq. (33)]. The PEA then returns one estimate with probability 1/2. If it is possible to prepare a single eigenstate of U (say that of eigenvalue $e^{+i\theta}$), $|\Psi_+\rangle$, then one can

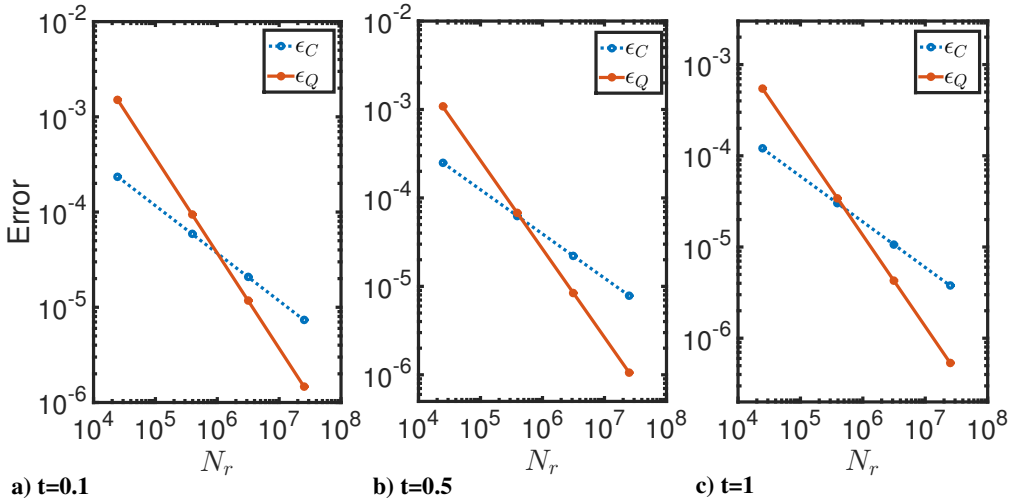


Fig. 6 Comparison of the errors output by the classical MC method (ϵ_C) and our quantum PEA (ϵ_Q). The results are for the 4-th central moment of the binary mixing model studied in Secs. III B and V C, for different values of t and N_r. The latter refers to the number of repetitions of the classical MC method or the number of state preparations needed by our quantum PEA. The data points are for $N_r = 2^m \times L$, where $m = 10, 14, 17, 20$ and $L = 24$. The confidence level of the estimation is 99.75%. The logarithmic scale allows us to observe clearly a better precision dependence, in terms of N_r, for our quantum PEA than for MC simulations.

use the results of [66] to parallelize the algorithm. To obtain a circuit of short depth, the steps to estimate the k-th bit of the phase need not be implemented sequentially. This can be overcome by first preparing the M ancillas in the cat state $(|00\ldots0\rangle + |11\ldots1\rangle)/\sqrt{2}$, rather than $|+\rangle^{\otimes m}$ as in the sequential approach, and by preparing M copies of $|\Psi_+\rangle$. (Recall that $M = 2^m$.) Then, the unitary operation U can be implemented in parallel conditional on the state of each of the ancillas. The final ancillary state contains information about $M\theta$ that can be extracted following the results of [6]. The quantum circuit is depicted in Fig. 7.

To prepare a single copy of the eigenstate $|\Psi_+\rangle$ from $|\Psi\rangle$, we will simulate a projective measurement of the eigenstates of U on $|\Psi\rangle$. This measurement can be made via the PEA of Fig. 3 using a number of bits of precision, m', that is sufficiently large to distinguish between the phases $+\theta$ and $-\theta$ with high probability. Then, $m' = O(|\log(\theta)|)$ and a single run of phase estimation requires using U order $1/\theta$ times. If we succeed in measuring $+\theta$, the quantum state $|\Psi\rangle$ is projected into

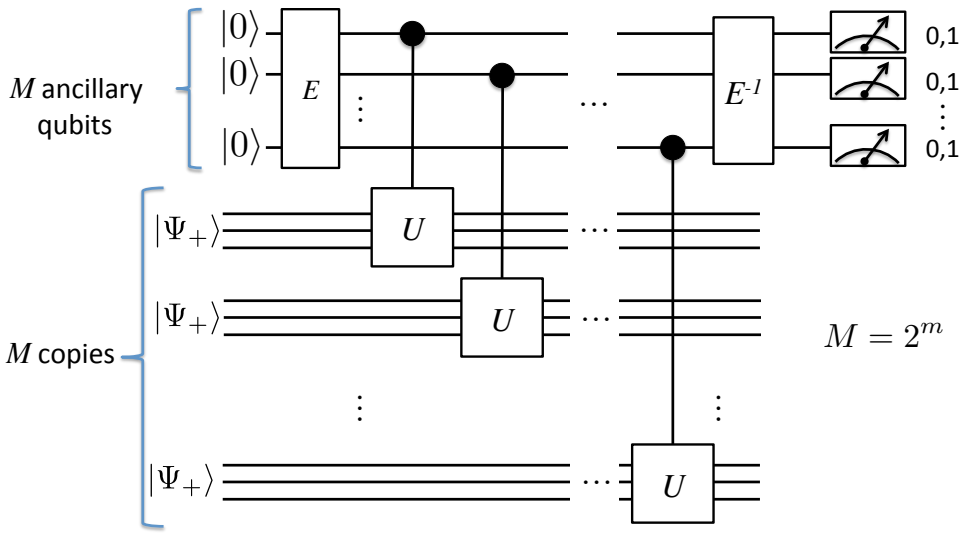


Fig. 7 The parallel version of the PEA. The algorithm outputs an estimate $\hat{\theta}$ of the eigenphase of the unitary U. The number of resources is almost linear in $1/\epsilon$, where ϵ is the precision of the estimation. E denotes the entangling gate that prepares the ancillary quantum state $(|00\ldots0\rangle + |11\ldots1\rangle)/\sqrt{2}$. The conditional W operations commute with each other and can be implemented in parallel. The j-th bit of the estimate $\hat{\theta}$ eigenphase is the outcome of measurement on 2^{j-1}-th qubit.

$|\Psi_+\rangle$ and we keep that copy. If we fail, we discard the state, prepare $|\Psi\rangle$, and run the PEA again. Since the probability of succeeding in the preparation of $|\Psi_+\rangle$ is exactly 1/2, the number of implementations of PEA to create M copies is of order $M \log M$. For constant θ, the overall number of resources of the parallel algorithm is almost linear in M, as in the sequential case.

VI. Conclusions and Outlook

In this paper, we have presented a quantum algorithm for a turbulent mixing problem, which provides a quadratic speedup over classical MC methods in terms of the number of repetitions that are required to achieve a given level of precision. We analyzed the application of our algorithm to a binary scalar mixing process modeled by means of the coalescence/dispersion (C/D) closure, estimating the precision obtained as a function of the number of repetitions for classical MC techniques and our quantum algorithm, obtaining the expected speedup. We also analyzed in which ways the quantum algorithm can be parallelized to restrict the number of resources used.

On its own, this algorithm can be applied to a range of turbulent mixing problems, and demonstrates the potential power of quantum computing in this area. More broadly, this first example study gives us a basis from which to further analyze questions associated with the potential applications of future quantum computers in fluid dynamics. Although we expect that it will be some time before the implementation of a quantum computer large enough to run this algorithm, recent developments in quantum hardware are very encouraging [11–18]. Beginning now to investigate the detailed application of this technology to computational problems in fluid dynamics should both motivate further developments in quantum computing, and help us to understand better the likely impact of this potentially disruptive technology. Our specific example highlights also how studying potential applications forces us to ask new questions about the implementation of quantum algorithms - in this case, especially regarding the parallelization of our procedure.

Acknowledgments

This work was supported by AFOSR grant FA9550-12-1-0057, Quantum Speedup for Turbulent Combustion Simulations, which brought together the authors from physics, quantum information and engineering. Results were obtained using the EPSRC funded ARCHIE-WeSt High Performance Computer (www.archie-west.ac.uk). EPSRC grant no. EP/K000586/1.

References

[1] Montanaro, A., "Quantum algorithms: an overview," *Npj Quantum Information*, Vol. 2, 01 2016, pp. 15023 EP –.

[2] Grover, L. K., "A fast quantum mechanical algorithm for database search," *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, 1996, pp. 212–219.

[3] Szegedy, M., "Quantum speed-up of Markov chain based algorithms," *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, 2004, pp. 32–41.

[4] Magniez, F., Nayak, A., Roland, J., and Santha, M., "Search via Quantum Walk," *Proceedings of the 39th Symposium on the Theory of Computing*, 2006, pp. 575–584.

[5] Giovannetti, V., Lloyd, S., and Maccone, L., "Quantum metrology," *Phys. Rev. Lett.*, Vol. 96, 2006, pp. 010401.

[6] Knill, E., Ortiz, G., and Somma, R., "Optimal Quantum Measurements of Expectation Values of Observables," *Phys. Rev. A*, Vol. 75, 2007, pp. 012328.

[7] Somma, R. D., Boixo, S., Barnum, H., and Knill, E., "Quantum Simulations of Classical Annealing Processes," *Phys. Rev. Lett.*, Vol. 101, 2008, pp. 130504–4.

[8] Poulin, D. and Wocjan, P., "Preparing Ground States of Quantum Many-Body Systems on a Quantum Computer," *Phys. Rev. Lett.*, Vol. 102, 2009, pp. 130503.

[9] Montanaro, A., "Quantum Speedup of Monte Carlo Methods," *Proc. Roy. Soc. Ser. A*, Vol. 471, 2015, pp. 20150301.

[10] Chowdhury, A. and Somma, R. D., "Quantum algorithms for Gibbs sampling and hitting-time estimation," *arXiv:1603.02940*, 2016.

[11] Kelly, J., Barends, R., Fowler, A. G., Megrant, A., Jeffrey, E., White, T. C., Sank, D., Mutus, J. Y., Campbell, B., Chen, Y., Chen, Z., Chiaro, B., Dunsworth, A., Hoi, I. C., Neill, C., O/'Malley, P. J. J., Quintana, C., Roushan, P., Vainsencher, A., Wenner, J., Cleland, A. N., and Martinis, J. M., "State preservation by repetitive error detection in a superconducting quantum circuit," *Nature*, Vol. 519, No. 7541, 03 2015, pp. 66–69.

[12] Martinez, E. A., Muschik, C. A., Schindler, P., Nigg, D., Erhard, A., Heyl, M., Hauke, P., Dalmonte, M., Monz, T., Zoller, P., and Blatt, R., "Real-time dynamics of lattice gauge theories with a few-qubit quantum computer," *Nature*, Vol. 534, No. 7608, 06 2016, pp. 516–519.

[13] Monz, T., Nigg, D., Martinez, E. A., Brandl, M. F., Schindler, P., Rines, R., Wang, S. X., Chuang, I. L., and Blatt, R., "Realization of a scalable Shor algorithm," *Science*, Vol. 351, No. 6277, 2016, pp. 1068–1070.

[14] Debnath, S., Linke, N. M., Figgatt, C., Landsman, K. A., Wright, K., and Monroe, C., "Demonstration of a small programmable quantum computer with atomic qubits," *Nature*, Vol. 536, No. 7614, 08 2016, pp. 63–66.

[15] Blatt, R. and Wineland, D., "Entangled states of trapped atomic ions," *Nature*, Vol. 453, No. 7198, 06 2008, pp. 1008–1015.

[16] Clarke, J. and Wilhelm, F. K., "Superconducting quantum bits," *Nature*, Vol. 453, No. 7198, 06 2008, pp. 1031–1042.

[17] Devoret, M. H. and Schoelkopf, R. J., "Superconducting Circuits for Quantum Information: An Outlook," *Science*, Vol. 339, No. 6124, 2013, pp. 1169–1174.

[18] Awschalom, D. D., Bassett, L. C., Dzurak, A. S., Hu, E. L., and Petta, J. R., "Quantum Spintronics: Engineering and Manipulating Atom-Like Spins in Semiconductors," *Science*, Vol. 339, No. 6124, 2013,

pp. 1174–1179.

[19] Toor, H. L., "Mass Transfer in Dilute Turbulent and Nonturbulent Systems with Rapid Irreversible Reactions and Equal Diffusivities," *AIChE J.*, Vol. 8, 1962, pp. 70–78.

[20] Brodkey, R. S., editor, *Turbulence in Mixing Operation*, Academic Press, New York, NY, 1975.

[21] Libby, P. A. and Williams, F. A., editors, *Turbulent Reacting Flows*, Vol. 44 of *Topics in Applied Physics*, Springer-Verlag, Heidelberg, 1980.

[22] Pope, S. B., "PDF Methods for Turbulent Reactive Flows," *Prog. Energ. Combust.*, Vol. 11, 1985, pp. 119–192.

[23] Givi, P., "Model Free Simulations of Turbulent Reactive Flows," *Prog. Energ. Combust.*, Vol. 15, 1989, pp. 1–107.

[24] Kollmann, W., "The PDF Approach to Turbulent Flow," *Theor. Comp. Fluid Dyn.*, Vol. 1, 1990, pp. 249–285.

[25] Libby, P. A. and Williams, F. A., editors, *Turbulent Reacting Flows*, Academic Press, London, England, 1994.

[26] Pope, S. B., *Turbulent Flows*, Cambridge University Press, Cambridge, UK, 2000.

[27] Haworth, D. C., "Progress in Probability Density Function Methods for Turbulent Reacting Flows," *Prog. Energ. Combust.*, Vol. 36, No. 2, 2010, pp. 168–259.

[28] Dopazo, C., "Recent Developments in PDF Methods," Libby and Williams [25], chap. 7, pp. 375–474.

[29] Fox, R. O., *Computational Models for Turbulent Reacting Flows*, Cambridge University Press, Cambridge, UK, 2003.

[30] Givi, P., "Filtered Density Function for Subgrid Scale Modeling of Turbulent Combustion," *AIAA J.*, Vol. 44, No. 1, 2006, pp. 16–23.

[31] Haworth, D. C. and Pope, S. B., "Transported Probability Density Function Methods for Reynolds-Averaged and Large-Eddy Simulations," *Turbulent Combustion Modeling*, edited by T. Echekki and E. Mastorakos, Vol. 95 of *Fluid Mechanics and Its Applications*, Springer Netherlands, 2011, pp. 119–142.

[32] Haworth, D. C. and Pope, S. B., "Monte Carlo Solutions of a Joint PDF Equation for Turbulent Flows in General Orthogonal Coordinates," *J. Comput. Phys.*, Vol. 72, No. 2, 1987, pp. 311–346.

[33] Kloeden, P. E., Platen, E., and Schurz, H., *Numerical Solution of Stochastic Differential Equations through Computer Experiments*, Springer-Verlag, New York, NY, corrected second printing ed., 1997.

[34] Madnia, C. K., Jaberi, F. A., and Givi, P., "Large Eddy Simulation of Heat and Mass Transport in Turbulent Flows," *Handbook of Numerical Heat Transfer*, edited by W. J. Minkowycz, E. M. Sparrow,

and J. Y. Murthy, chap. 5, John Wiley & Sons, Inc., New York, NY, 2nd ed., 2006, pp. 167–189.

[35] Giovannetti, V., Lloyd, S., and Maccone, L., "Quantum-enhanced measurements: beating the standard quantum limit," *Science*, Vol. 306, 2004, pp. 1330.

[36] Janicka, J., Kolbe, W., and Kollmann, W., "Closure of the Transport Equation for the Probability Density Function of Turbulent Scalar Field," *J. Non-Equil. Thermodyn.*, Vol. 4, 1979, pp. 47–66.

[37] Pope, S. B., "An Improved Turbulent Mixing Model," *Combust. Sci. Technol.*, Vol. 28, 1982, pp. 131–145.

[38] Kosály, G. and Givi, P., "Modeling of Turbulent Molecular Mixing," *Combust. Flame*, Vol. 70, 1987, pp. 101–118.

[39] O'Brien, E. E., "The Probability Density Function (PDF) Approach to Reacting Turbulent Flows," Libby and Williams [21], chap. 5, pp. 185–218.

[40] Pope, S. B., "The Probability Approach to Modeling of Turbulent Reacting Flows," *Combust. Flame*, Vol. 27, 1976, pp. 299–312.

[41] Chen, H., Chen, S., and Kraichnan, R. H., "Probability Distribution of a Stochastically Advected Scalar Field," *Phys. Rev. Lett.*, Vol. 63, No. 24, 1989, pp. 2657–2660.

[42] Pope, S. B., "Mapping Closures for Turbulent Mixing and Reaction," *Theor. Comp. Fluid Dyn.*, Vol. 2, 1991, pp. 255–270.

[43] Valiño, L. and Dopazo, C., "A Binomial Langevin Model for Turbulent Mixing," *Phys. Fluids A*, Vol. 3, No. 12, 1991, pp. 3034–3037.

[44] Miller, R. S., Frankel, S. H., Madnia, C. K., and Givi, P., "Johnson-Edgeworth Translation for Probability Modeling of Binary Scalar Mixing in Turbulent Flows," *Combust. Sci. Technol.*, Vol. 91, No. 1-3, 1993, pp. 21–52.

[45] Subramaniam, S. and Pope, S. B., "A Mixing Model for Turbulent Reactive Flows Based on Euclidean Minimum Spanning Trees," *Combust. Flame*, Vol. 115, 1998, pp. 487–514.

[46] Klimenko, A. Y. and Pope, S. B., "The Modeling of Turbulent Reactive Flows Based on Multiple Mapping Conditioning," *Phys. Fluids*, Vol. 15, No. 7, 2003, pp. 1907–1925.

[47] Pope, S. B., "A Model for Turbulent Mixing Based on Shadow-Position Conditioning," *Phys. Fluids*, Vol. 25, No. 11, 2013, pp. 110803.

[48] Jaberi, F. A., Miller, R. S., Madnia, C. K., and Givi, P., "Non-Gaussian Scalar Statistics in Homogeneous Turbulence," *J. Fluid Mech.*, Vol. 313, 1996, pp. 241–282.

[49] Pope, S. B., "Small Scales, Many Species and the Manifold Challenges of Turbulent Combustion," *Proc. Combust. Inst.*, Vol. 34, No. 1, 2013, pp. 1–31.

[50] Curl, R. L., "Dispersed Phase Mixing: I. Theory and Effects in Simple Reactors," *AIChE J.*, Vol. 9, No. 2, 1963, pp. 175–181.

[51] Borghi, R., "Turbulent Combustion Modeling," *Prog. Energ. Combust.*, Vol. 14, 1988, pp. 245–292.

[52] Hoeffding, W., "Probability Inequalities for Sums of Bounded Random Variables," *J. Am. Stat. Assoc.*, Vol. 58, 1963, pp. 13–30.

[53] Nielsen, M. A. and Chuang, I. L., *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, UK, 2000.

[54] Kaye, P., Laflamme, R., and Mosca, M., *An Introduction to Quantum Computing*, Oxford University Press, USA, 2007.

[55] Dirac, P., "A new notation for quantum mechanics," *Math. Proc. Cambridge Phil. Soc.*, Vol. 35, 1939, pp. 416–418.

[56] Griffiths, D. J., *Introduction to Quantum Mechanics*, Pearson, UK, 2014.

[57] Kitaev, A. Y., "Quantum measurements and the Abelian Stabilizer Problem," *arxiv:quant-ph/9511026*, Nov. 1995.

[58] Kitaev, A. Y., Shen, A., and Vyalyi, M., *Classical and Quantum Computation*, American Mathematical Society, 2002.

[59] Nawrocki, W., *Introduction to Quantum Metrology*, Springer, 2015.

[60] Cleve, R., Ekert, A., Macchiavello, C., and Mosca, M., "Quantum algorithms revisited," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Vol. 454, No. 1969, 1998, pp. 339–354.

[61] Griffiths, R. and Niu, C.-S., "Semiclassical Fourier Transform for Quantum Computation," *Phys. Rev. Lett.*, Vol. 76, 1996, pp. 3228.

[62] Brassard, G., Høyer, P., Mosca, M., and Tapp, A., *Quantum Computation and Quantum Information: A Millennium Volume*, AMS Contemporary Mathematics Series, Am. Math. Soc., USA, 2000.

[63] Nagaj, D., Wocjan, P., and Zhang, Y., "Fast Amplification of QMA," *Quant. Inf. Comp.*, Vol. 9, April 2009, pp. 1053.

[64] Berry, D., Childs, A., Cleve, R., Kothari, R., and Somma, R., "Simulating Hamiltonian Dynamics with a Truncated Taylor Series," *Phys. Rev. Lett.*, Vol. 114, 2015, pp. 090502.

[65] Childs, A., Kothari, R., and Somma, R. D., "Quantum linear systems algorithm with exponentially improved dependence on precision," *arXiv:1511.02306*, 2015.

[66] Bollinger, J. J., Itano, W. M., Wineland, D. J., and Heinzen, D. J., "Optimal frequency measurements with maximally correlated states," *Phys. Rev. A*, Vol. 54, 1996, pp. R4649.