Exponential quantum speedup in simulating coupled classical oscillators

Ryan Babbush,[1] Dominic W. Berry,[2] Robin Kothari,[1] Rolando D. Somma,[1] and Nathan Wiebe[3, 4, 5]

[1] *Google Quantum AI, Venice, CA, United States*
[2] *School of Mathematical and Physical Sciences,*
Macquarie University, Sydney, NSW, Australia
[3] *Department of Computer Science, University of Toronto, Toronto, ON, Canada*
[4] *Pacific Northwest National Laboratory, Richland, WA, United States*
[5] *Canadian Institute for Advanced Research, Toronto, ON, Canada*
(Dated: September 21, 2023)

We present a quantum algorithm for simulating the classical dynamics of 2^n coupled oscillators (e.g., 2^n masses coupled by springs). Our approach leverages a mapping between the Schrödinger equation and Newton's equation for harmonic potentials such that the amplitudes of the evolved quantum state encode the momenta and displacements of the classical oscillators. When individual masses and spring constants can be efficiently queried, and when the initial state can be efficiently prepared, the complexity of our quantum algorithm is polynomial in n, almost linear in the evolution time, and sublinear in the sparsity. As an example application, we apply our quantum algorithm to efficiently estimate the kinetic energy of an oscillator at any time. We show that any classical algorithm solving this same problem is inefficient and must make $2^{\Omega(n)}$ queries to the oracle and, when the oracles are instantiated by efficient quantum circuits, the problem is BQP-complete. Thus, our approach solves a potentially practical application with an exponential speedup over classical computers. Finally, we show that under similar conditions our approach can efficiently simulate more general classical harmonic systems with 2^n modes.

I. INTRODUCTION

The efficient simulation of quantum dynamics is among the most promising areas of quantum computing [1]. This is due to having practical applications as well as universality providing strong complexity theoretic evidence for exponential quantum advantage [2, 3]. Several other applications with similarly favorable qualities are known (e.g., factoring [4]) but the discovery of more compelling use cases remains critical for understanding and motivating the value proposition of quantum computers.

One approach for leveraging the power of Hamiltonian simulation would be to consider the space of problems that reduces to it. Hamiltonian dynamics is an example of a homogeneous, first order partial differential equation that gives rise to unitary evolutions. Thus, it is natural to explore whether other differential equations can be mapped to Hamiltonian simulation. The goal would be a more natural (albeit perhaps more narrow) alternative to solving differential equations compared with methods [5–7] leveraging quantum linear systems algorithms [8–10]. Past work has sought to develop Hamiltonian simulation approaches for a limited set of other differential equations but thus far, has only succeeded in obtaining polynomial speedups [11–13].

In this paper, we discuss a rich set of problems in *classical* dynamics that can be mapped to Hamiltonian simulation and solved with exponential quantum advantage. As a prominent example, our approach can simulate the dynamics of exponentially many coupled classical oscillators in polynomial time. Such systems describe a variety of physical phenomena including: networks of masses and springs [14], circuits with capacitors and inductors [15], models of neuron activity [16], and vibrations in molecules [17], materials, and mechanical structures. More generally, the harmonic approximation (defined by a quadratic potential) arises as the first order correction to equilibrium in bound systems.

That the dynamics of coupled classical oscillators can be studied via Hamiltonian simulation is perhaps not too surprising. Solutions to the Schrödinger equation contain oscillatory terms and interference, and we are effectively using these properties to simulate the same phenomena in the classical system. This correspondence was also presaged by the finding that Grover search can be implemented (in the absence of errors) using mechanical wave interference [18]. Similarly, the problem of simulating the (discrete) classical wave equation is a special case of the oscillator dynamics problem considered here, and a quantum algorithm to solve the classical wave equation via Hamiltonian simulation has been studied in Ref. [11]. However, despite the seemingly natural connection, our mapping involves a number of subtle technical features and modern quantum algorithms are required to efficiently simulate the resultant Hamiltonians.

Besides providing a classical-to-quantum reduction, another key to obtain exponential quantum advantage for this problem is the way we encode physical quantities in quantum states. In our case, this encoding is motivated by energy conservation (being different to the one in Ref. [11]). Specifically, we encode quantities related to the displacements and momenta of the classical system in the amplitudes of a quantum state. Thus, extracting the full configuration of the classical system would scale polynomially in the Hilbert space dimension, precluding a large quantum speedup. Nevertheless, interesting global properties, like estimating the kinetic or potential energies at any time, can still be computed efficiently. Likewise, our methods only provide an exponential speedup for the dynamics of sparsely coupled oscillator networks when masses, spring constants, and initial states can be efficiently computed. Fortunately, many interesting systems meet those conditions.

Finally, we provide strong evidence that such simulations cannot also be performed efficiently on a classical computer. In particular, we are able to show that estimating the kinetic energy of simple instances of these systems at any time that is $\mathrm{poly}(n)$ requires $2^{\Omega(n)}$ queries to the oracles in the worst case and, when the oracles are instantiated by quantum circuits of $\mathrm{poly}(n)$ size, the problem is BQP-complete. Thus, all problems efficiently solved by a quantum computer can be reduced to an instance of simulating exponentially many coupled classical oscillators, which can also be solved efficiently by our approach.

The rest of the paper is organized as follows. In Sec. II we describe classical systems of oscillators, define the associated simulation problems of interest, and state our main results. In Sec. III we show how these problems reduce to instances of Hamiltonian simulation, giving rise to an efficient quantum algorithm. We discuss some applications of this algorithm in Sec. V, making emphasis on the problem of estimating the time-dependent kinetic (or potential) energies of the oscillators. In Sec. VI A we show the exponential lower bound for classical algorithms for this problem in the oracle setting, and in Sec. VI B we show that this problem is BQP-complete when the oracles can be accessed via efficient quantum circuits. Finally, in Sec. VII we generalize our approach to efficiently simulating classical systems under the harmonic approximation. Detailed proofs of our claims are provided in the Appendix. We also provide a comparison of our results with those of related work on quantum algorithms for differential equations in Appendix G.
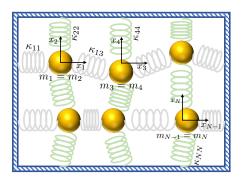

FIG. 1. An example system of $N/2$ oscillators in $D = 2$ dimensions. We use $x_1(t)$ for the first coordinate of the first mass and $x_2(t)$ for the second coordinate of the first mass, and so on. Since the first and second mass now correspond to the same original mass, $m_1 = m_2$.

II. SIMULATING COUPLED OSCILLATORS: MAIN RESULTS

We consider a classical system of coupled harmonic oscillators, i.e., $N = 2^n$ point (positive) masses $m_1, \ldots, m_N$ that are coupled with each other by springs. At any time $t \geq 0$, the displacements (with respect to their rest position) and velocities of the masses are given by $\vec{x}(t) = (x_1(t), \ldots, x_N(t))^T \in \mathbb{R}^N$ and $\dot{\vec{x}}(t) = (\dot{x}_1(t), \ldots, \dot{x}_N(t))^T \in \mathbb{R}^N$, respectively, where $\dot{a}(t) = \frac{d}{dt}a(t)$ and $\ddot{a}(t) = \frac{d^2}{dt^2}a(t)$. We let $\kappa_{jk} = \kappa_{kj} \geq 0$ be the spring constants that couple the jth and k^{th} oscillator, and $\kappa_{jj} \geq 0$ is the spring constant that connects the jth oscillator to a "wall". We have described it for dimension $D = 1$ for simplicity (i.e., we assigned one spatial coordinate to each oscillator), but the same formulation holds for coupled harmonic oscillators in D spatial dimensions, for arbitrary D, by using D coordinates to represent the position of a single oscillator (see Fig. 1 for an example).

In the harmonic approximation, the dynamics of the oscillators can be determined from the initial values $\vec{x}(0)$ and $\dot{\vec{x}}(0)$ and Newton's equation (for all $j \in [N] := \{1, \ldots, N\}$):

$$m_j \ddot{x}_j(t) = \sum_{k \neq j} \kappa_{jk}\big(x_k(t) - x_j(t)\big) - \kappa_{jj} x_j(t) \,. \quad (1)$$

In matrix form, this is $\mathbf{M}\ddot{\vec{x}}(t) = -\mathbf{F}\vec{x}(t)$, where $\mathbf{M}$ is a $N \times N$ diagonal matrix with entries $m_j > 0$ and $\mathbf{F}$ is the $N \times N$ matrix whose diagonal and off-diagonal entries are $f_{jj} = \sum_k \kappa_{jk}$ and $f_{jk} = -\kappa_{jk}$, respectively. (Observe that $\mathbf{F}$ is the discrete Laplacian of a weighted graph.) Solutions to Eq. (1) are

well understood and can be expressed in terms of normal modes [14], which is essentially a way of describing the system as N uncoupled harmonic oscillators in a different basis. Classical algorithms that compute $\vec{x}(t)$ and $\dot{\vec{x}}(t)$ have complexity poly(N) or, equivalently, exp(n). (In this paper we use exp(m), poly(m) and polylog(m) to mean $\mathcal{O}(k^m)$, $\mathcal{O}(m^k)$, and $\mathcal{O}(\log^k m)$ respectively, for some constant k. (We use "log" to mean logarithm to base 2.)

Our goal is to provide a quantum algorithm for simulating the dynamics of the classical oscillators efficiently, in time poly(n). Doing so requires a particular notion of "simulation", since any algorithm that outputs the full vectors $\vec{x}(t)$ or $\dot{\vec{x}}(t)$ would necessarily have complexity at least linear in N. Specifically, we consider a problem formulation where the output is a quantum state with some amplitudes proportional to $\sqrt{m_j}\dot{x}_j(t)$ and others to $\sqrt{\kappa_{jk}}(x_j(t) - x_k(t))$ or $\sqrt{\kappa_{jj}}x_j(t)$, and the input masses and spring constants are provided through oracles in the usual way; see App. A. (Here and throughout this paper, $\sqrt{x}$ is the principal square root of $x \geq 0$ and $\sqrt{\mathbf{X}}$ is the principal square root of a positive semidefinite matrix $\mathbf{X} \succcurlyeq 0$.) The formal problem is as follows:

Problem 1. *Let $\mathbf{K}$ be the $N \times N$ symmetric matrix of spring constants $\kappa_{jk} \geq 0$ and assume it is d-sparse (i.e., there are at most d non-zero entries in each row). Let $\mathbf{M}$ be the $N \times N$ diagonal matrix of masses $m_j > 0$ and define the normalized state*

$$|\psi(t)\rangle := \frac{1}{\sqrt{2E}}\begin{pmatrix} \sqrt{\mathbf{M}}\dot{\vec{x}}(t) \\ i\vec{\mu}(t) \end{pmatrix} , \qquad (2)$$

where $E > 0$ is a constant, and $\vec{\mu}(t) \in \mathbb{R}^M$ ($M := N(N+1)/2$) is a vector with N entries $\sqrt{\kappa_{jj}}x_j(t)$ and $N(N-1)/2$ entries $\sqrt{\kappa_{jk}}(x_j(t) - x_k(t))$, with $k > j$. Assume we are given oracle access to $\mathbf{K}$ and $\mathbf{M}$, and oracle access to a unitary $\mathcal{W}$ that prepares the initial state, i.e., $\mathcal{W}|0\rangle \mapsto |\psi(0)\rangle$. Given $t \geq 0$ and $\epsilon > 0$, the goal is to output a state that is ϵ-close to $|\psi(t)\rangle$ in Euclidean norm.

Our main result is a quantum algorithm that prepares $|\psi(t)\rangle$ efficiently:

Theorem 1. *Problem 1 can be solved with a quantum algorithm that makes $Q = \mathcal{O}(\tau + \log(1/\epsilon))$ queries to the oracles for $\mathbf{K}$ and $\mathbf{M}$, uses*

$$G = \mathcal{O}\left(Q \times \log^2\left(\frac{N\tau}{\epsilon} \frac{m_{\max}}{m_{\min}} \right) \right) \qquad (3)$$

two-qubit gates, and uses $\mathcal{W}$ once, where $\tau := t\sqrt{\aleph d} \geq 1$, $\aleph := \kappa_{\max}/m_{\min}$, $m_{\max} \geq m_j \geq m_{\min} > 0$ and $\kappa_{\max} \geq \kappa_{jk}$ for all $j, k \in [N]$ are known quantities.

In the $\mathcal{O}$ notation above, the asymptotically large parameters are $N, \tau, 1/\epsilon$, and $m_{\max}/m_{\min}$. This complexity has explicit dependence on $n = \log(N)$ that is polynomial. If τ, $\log(1/\epsilon)$, and $\log(m_{\max}/m_{\min})$ are poly(n), and all oracles (including $\mathcal{W}$) can be performed with cost poly(n), then the entire algorithm has complexity poly(n). An interesting feature of this complexity is that it scales as the square root of the sparsity d, whereas Hamiltonian simulation complexities are usually linear in d. The complexity is also linear in τ, which bounds the maximum number of oscillations of the normal modes after time t.

Our algorithm can be used to determine properties of the system at time t. For example, the state $|\psi(t)\rangle$ encodes the velocities and displacements of the oscillators in a way that makes it easy for the estimation of the kinetic or potential energies, as we discuss below. We discuss in App. F that other encodings are also possible, and the choice of encoding may determine which initial states and properties (observables) are efficient to prepare and measure. Nevertheless, our main goal is to establish results for this particular encoding.

The constant in Eq. (2) is $E = K(t) + U(t)$, where $K(t) = \frac{1}{2}\dot{\vec{x}}(t)^T \mathbf{M}\dot{\vec{x}}(t)$ is the kinetic energy, i.e.,

$$K(t) = \frac{1}{2}\sum_j m_j \dot{x}_j(t)^2 , \qquad (4)$$

and $U(t) = \frac{1}{2}\vec{\mu}(t)^T \vec{\mu}(t)$ is the potential energy, i.e.,

$$U(t) = \frac{1}{2}\left(\sum_j \kappa_{jj}x_j(t)^2 + \sum_{k>j} \kappa_{jk}\big(x_j(t) - x_k(t)\big)^2 \right). \qquad (5)$$

Hence, E is the total energy of the system, which is time-independent as the system is closed. The support of $|\psi(t)\rangle$ on the subspace spanned by the first N basis vectors is $K(t)/E$, which can be estimated via a simple measurement on $|\psi(t)\rangle$; see Sec. V. The support on the other subspace is $U(t)/E$. As an example of the type of problem that can be solved by preparing $|\psi(t)\rangle$, consider the following:

Problem 2. *Given the same inputs as Problem 1 and an oracle for $V \subseteq [N]$, which is a subset of the oscillators, output an estimate $\hat{k}_V(t) \in \mathbb{R}$ such that*

$$|\hat{k}_V(t) - K_V(t)/E| \leq \epsilon , \qquad (6)$$

where $K_V(t) := \frac{1}{2}\sum_{j \in V} m_j \dot{x}_j(t)^2$ is the kinetic energy of V at time t.

In Sec. V, we prove that our quantum algorithm solves this problem with high probability with $\mathcal{O}(1/\epsilon)$

3

uses of the quantum algorithm from Thm. 1. We also show a related result for the estimation of the potential energy stored on a subset $V \subseteq [N] \times [N]$ of springs at time t.

Problems 1 and 2 are formulated using a specific input and output format, and one might wonder if a classical algorithm could also solve these problems efficiently. The answer is no: we show that any classical algorithm solving Problem 2 must make $\text{poly}(N)$ or $\exp(n)$ queries to the oracles in general.

Theorem 2. *Any classical algorithm that solves Problem 2 with high probability must make $2^{\Omega(n)}$ queries to the oracle for $\mathbf{K}$. This lower bound holds even if we further require that all $m_j = 1$, all $\kappa_{jk} \in \{0, 1\}$, $d \le 3$, and the initial state is $|\psi(0)\rangle = |0\rangle^{\otimes q}$ for some $q = \text{poly}(n)$, which corresponds to $\vec{x}(0) = (0, 0, \dots, 0)^T$ and $\dot{\vec{x}}(0) = (1, 0, \dots, 0)^T$.*

To provide more evidence of the impossibility of efficient classical simulations and prove a stronger hardness result, we define a non-oracular version of Problem 2 and show that it is BQP-complete. Thus, no classical algorithm can solve this non-oracular problem efficiently (in time $\text{poly}(n)$) unless it can also solve every problem solved by quantum computers in polynomial time. The same observation essentially applies to Problem 2 as a result. Formally, to be BQP-complete a problem must be a decision problem (i.e., a problem where there are only two possible answers), so we define a decision version of Problem 2 with $|V| = 1$ and a sparse initial state:

Problem 3. *The setup is as in Problem 2, but we are given efficient quantum circuits (i.e., circuits with $\text{poly}(n)$ gates) to implement the oracles for $\mathbf{K}$ and $\mathbf{M}$, an explicit description of $|\psi(0)\rangle$, which is required to have a constant number of nonzero entries, and a label $v \in [N]$ of a single oscillator. We additionally require that τ, $1/\epsilon$, and $m_{\max}/m_{\min}$ are bounded by $\text{poly}(n)$. The problem is to decide if $K_v(t)/E$, the kinetic energy of oscillator v as a fraction of total energy, is at least $1/\text{poly}(n)$ or at most $1/\exp(\sqrt{n})$, promised that one of these holds.*

Note that this problem has an input of size $\text{poly}(n)$ and our algorithm for Problem 1 will solve this on a quantum computer in time $\text{poly}(n)$. We then show this problem is BQP-complete in Sec. VI B:

Theorem 3. *Problem 3 is BQP-complete. The problem remains BQP-complete even if we further require that all $\kappa_{jk} \le 4$, $m_j = 1$, $d \le 4$, and the initial state is $|\psi(0)\rangle = |0\rangle^{\otimes q} \otimes |-\rangle$ for some $q = \text{poly}(n)$, which corresponds to $\vec{x}(0) = (0, 0, \dots, 0)^T$ and $\dot{\vec{x}}(0) = (+1, -1, 0, \dots, 0)^T$.*

Finally, we show that our results can be extended to address more general classical systems that arise naturally under the harmonic approximation [14]. In these cases, for example, the matrix $\mathbf{M}$ resulting from Eq. 1 is not necessarily diagonal and the off-diagonal entries of $\mathbf{F}$ are not necessarily non-positive. A simple change of variables takes Eq. 1 to the standard form $\ddot{\vec{y}}(t) = -\mathbf{A}\vec{y}(t)$, where $\vec{y}(t) = \sqrt{\mathbf{M}}\vec{q}(t)$, and $\vec{q}(t) \in \mathbb{R}^N$ are the generalized displacements. We then consider the following generalization of Problem 1:

Problem 4. *Let $\mathbf{A} \succcurlyeq 0$ be an $N \times N$ real-symmetric, PSD, d-sparse matrix. Define the normalized state*

$$|\psi(t)\rangle := \frac{1}{\sqrt{2E}} \begin{pmatrix} \dot{\vec{y}}(t) \\ i\vec{\mu}(t) \end{pmatrix} , \qquad (7)$$

where $E > 0$ is a constant and $\vec{\mu}(t) := \sqrt{\mathbf{A}}\vec{y}(t) \in \mathbb{C}^N$. Assume we are given oracle access to $\mathbf{A}$ and oracle access to a unitary $\mathcal{W}$ that prepares the initial state $|\psi(0)\rangle$. Given t and ϵ, the goal is to output a state that is ϵ-close to $|\psi(t)\rangle$ in Euclidean norm.

The constant E in Eq. (7) is also the energy of the system. The encoding is different from the one used for Problem 1; for example $\sqrt{\mathbf{A}}$ is of dimension $N \times N$ and might not be sparse even if $\mathbf{A}$ is. However, the support of $|\psi(t)\rangle$ on the subspace spanned by the first N basis states is still $K(t)/E$, where $K(t)$ is the kinetic energy. We then show the following:

Theorem 4. *Problem 4 can be solved with a quantum algorithm that makes*

$$Q = \mathcal{O}\left(\|\mathbf{A}\|_{\max} d \log(1/\epsilon) \min\left(\frac{t\sqrt{\|\mathbf{A}^{-1}\|}}{\epsilon}, \frac{t^2}{\epsilon^2} \right) \right) \qquad (8)$$

queries to the oracles, uses $\widetilde{\mathcal{O}}(Q \times \text{polylog}(N/\epsilon))$ two-qubit gates, and uses $\mathcal{W}$ once.

The $\tilde{\mathcal{O}}$ notation hides logarithmic factors in several parameters that specify the input.

III. REDUCTION TO QUANTUM EVOLUTION

We show how to reduce Problem 1 to time evolution of a quantum system. A change of variables where $\vec{y}(t) := \sqrt{\mathbf{M}}\vec{x}(t)$ allows us to write Eq. (1) as

$$\ddot{\vec{y}}(t) = -\mathbf{A}\vec{y}(t), \qquad (9)$$

where $\mathbf{A} := \sqrt{\mathbf{M}}^{-1} \mathbf{F} \sqrt{\mathbf{M}}^{-1} \succcurlyeq 0$ is PSD and real-symmetric, and $\mathbf{M} \succ 0$ is the diagonal matrix with

entries m_j. Any solution to Eq. (9) satisfies

$$\ddot{\vec{y}}(t) + i\sqrt{\mathbf{A}}\dot{\vec{y}}(t) = i\sqrt{\mathbf{A}}\Big(\dot{\vec{y}}(t) + i\sqrt{\mathbf{A}}\vec{y}(t)\Big) . \quad (10)$$

This is simply Schrödinger's equation induced by the Hamiltonian $-\sqrt{\mathbf{A}}$. Hence its solution is

$$\dot{\vec{y}}(t) + i\sqrt{\mathbf{A}}\vec{y}(t) = e^{it\sqrt{\mathbf{A}}}\Big(\dot{\vec{y}}(0) + i\sqrt{\mathbf{A}}\vec{y}(0)\Big). \quad (11)$$

Unfortunately, we do not have direct access to $-\sqrt{\mathbf{A}}$ and casting the problem as quantum evolution with simpler Hamiltonians requires additional steps. It is possible, however, to do Hamiltonian simulation with $-\sqrt{\mathbf{A}}$ given oracle access to $\mathbf{A}$ using quantum phase estimation, as we describe in Sec. VII, but that is less efficient than what we describe here. The phase estimation approach also gives rise to a different encoding, and some properties of the system are not as easy to access. For example, in the encoding used in Problem 1, the support of $|\psi(t)\rangle$ in basis states are either proportional to the kinetic energies or potential energies of the springs, but this might not be the case in other encodings.

Let $\mathbf{B}$ be any $N \times M$ matrix that satisfies $\mathbf{B}\mathbf{B}^\dagger = \mathbf{A}$ and $\mathbf{H}$ be the block Hamiltonian:

$$\mathbf{H} := -\begin{pmatrix} \mathbf{0} & \mathbf{B} \\ \mathbf{B}^\dagger & \mathbf{0} \end{pmatrix} , \quad (12)$$

where $\mathbf{0}$ are matrices of all zeroes. (The dimension of each $\mathbf{0}$ is clear from context.) Note that $\mathbf{H}$ acts on the space $\mathbb{C}^{N+M}$ and functions as the "square root" of $\mathbf{A}$, since the first block of $\mathbf{H}^2$ is $\mathbf{A}$. Schrödinger's equation induced by $\mathbf{H}$ is

$$|\dot{\psi}(t)\rangle = -i\mathbf{H}|\psi(t)\rangle , \quad (13)$$

where $|\psi(t)\rangle \in \mathbb{C}^{N+M}$ is the state of a quantum system at time t.

It can be verified by direct substitution that

$$|\psi(t)\rangle \propto \begin{pmatrix} \dot{\vec{y}}(t) \\ i\mathbf{B}^\dagger\vec{y}(t) \end{pmatrix} . \quad (14)$$

is a valid solution to Eq. (13). Hence we have

$$\begin{pmatrix} \dot{\vec{y}}(t) \\ i\mathbf{B}^\dagger\vec{y}(t) \end{pmatrix} = e^{-it\mathbf{H}} \begin{pmatrix} \dot{\vec{y}}(0) \\ i\mathbf{B}^\dagger\vec{y}(0) \end{pmatrix}, \quad (15)$$

which lets us compute $\dot{\vec{y}}(t)$ and $\mathbf{B}^\dagger\vec{y}(t)$ using Hamiltonian simulation. This generalizes Eq. (11) since $\mathbf{B}$ is not necessarily $-\sqrt{\mathbf{A}}$. (This formulation also stores the positions and velocities in different components of the vector, but this is a minor difference.)

To match the state representation of Problem 1, we need to choose $\mathbf{B}$ such that $\vec{\mu}(t) = \mathbf{B}^\dagger\vec{y}(t) = \mathbf{B}^\dagger\sqrt{\mathbf{M}}\vec{x}(t)$, where $\vec{\mu}(t) \in \mathbb{C}^M$ (for $M = N(N+1)/2$). We can express $\vec{\mu}(t)$ in the basis $\{|j,k\rangle : j \leq k \in [N]\}$, which is of size M. Since $\mathbf{F}$ is a graph Laplacian, we can obtain $\mathbf{B}$ from the incidence matrix of $\mathbf{F}$ given by

$$\sqrt{\mathbf{M}}\mathbf{B}|j,k\rangle = \begin{cases} \sqrt{\kappa_{jj}}|j\rangle , & \text{if } j = k \\ \sqrt{\kappa_{jk}}(|j\rangle - |k\rangle), & \text{if } j < k \end{cases} . \quad (16)$$

This choice satisfies $\sqrt{\mathbf{M}}\mathbf{B}(\sqrt{\mathbf{M}}\mathbf{B})^\dagger = \mathbf{F}$, and hence $\mathbf{B}\mathbf{B}^\dagger = \mathbf{A}$, because we can write $\mathbf{F} = \sqrt{\mathbf{M}}(\sum_{j \leq k} \mathbf{B}|j,k\rangle\langle j,k|\mathbf{B}^\dagger)\sqrt{\mathbf{M}}$ and each term in this sum describes the interaction between the jth and k^{th} oscillators:

$$\sqrt{\mathbf{M}}\mathbf{B}|j,j\rangle\langle j,j|\mathbf{B}^\dagger\sqrt{\mathbf{M}} = \kappa_{jj}|j\rangle\langle j| , \quad (17)$$

and for $j < k$,

$$\sqrt{\mathbf{M}}\mathbf{B}|j,k\rangle\langle j,k|\mathbf{B}^\dagger\sqrt{\mathbf{M}} = \kappa_{jk}(|j\rangle\langle j| + |k\rangle\langle k| \\ - |j\rangle\langle k| - |k\rangle\langle j|). \quad (18)$$

These reproduce $\mathbf{F}$ once we perform the sum. To obtain the action of $\mathbf{B}$ we apply $\sqrt{\mathbf{M}}^{-1}$ to Eq. (16).

Note that a similar approach based on the incidence matrix was taken in Ref. [11] to provide a quantum algorithm that simulates the wave equation. The setting considered in that paper corresponds to the special case of our problem where the graph is spatially local, and the masses and spring constants are uniform.

IV. QUANTUM ALGORITHM

Our quantum algorithm solves Problem 1 by preparing $|\psi(0)\rangle$ and simulating $\mathbf{H}$ for time t.

Access model: The Hamiltonian $\mathbf{H}$ is built from the matrix of spring constants $\mathbf{K}$ (or $\mathbf{F}$) and $\mathbf{M}$. As we wish to avoid complexities that are polynomial in N, we require succinct representations of these matrices. For maximum generality, we assume that access to $\mathbf{M}$ and the d-sparse $\mathbf{K}$ is provided by a black box, similar to that used in prior quantum algorithms [19, 20]. The black box is a unitary $\mathcal{S}$ that computes the masses m_j on input j and the nonzero entries of $\mathbf{K}$, i.e. κ_{jk}, on input (j,k), as well as their locations. This black box readily gives query access to $\mathbf{H}$ in Eq. (12), which is also d-sparse. Its entries are $\pm\sqrt{\kappa_{jk}/m_j}$ and these factors can be applied using the values of m_j and κ_{jk}; see App. A.

Simulation of H: Time evolution with the sparse **H** can be simulated using one of the methods in Refs. [21–24], which apply an approximation of the exponential $e^{-it\mathbf{H}}$. Such methods are tailored to our access model and achieve almost optimal scaling in the parameters ϵ, d, t, and $\|\mathbf{H}\|_{\max}$, the largest entry of **H** in absolute value. Here $\|\mathbf{H}\|_{\max}$ is at most $\sqrt{\aleph}$, which is defined in Thm. 1.

Complexity: The complexity of our approach is determined by the simulation of **H** for time t from the initial state. We consider the query complexity Q of the number of calls to $\mathcal{S}$, and we allow inverses and controlled forms of these oracles. This query complexity can also be taken to include the unitary for preparing the initial state, but only one call to that preparation is needed. We also consider the gate complexity G, which is the number of additional elementary gates, which could, for example, be single-qubit gates and CNOTs. We describe these as "two-qubit gates" in Thm. 1 for simplicity.

Using Ref. [23], the Hamiltonian evolution may be simulated with

$$\mathcal{O}(t\Lambda + \log(1/\epsilon)) \qquad (19)$$

calls to a block encoding of **H**, where the block encoding gives a factor of $1/\Lambda$. In App. A we show how to block encode **H** with $\Lambda = \sqrt{2\aleph d}$ using $\mathcal{O}(1)$ calls to $\mathcal{S}$, and so Eq. (19) gives the total query complexity Q given in Thm. 1 since $\tau := t\sqrt{\aleph d}$. The reason for the square root dependence on the sparsity is then because the algorithm involves simulation of **H**, which is effectively the square root of the d-sparse operator **A**. Note that $\aleph d$ is an upper bound on $\|\mathbf{A}\|$, implying that $\sqrt{\aleph d}$ is an upper bound on the largest frequency of the normal modes. See App. A for a more technical explanation.

To achieve final error ϵ, each block encoding should be given within error

$$\epsilon' = \mathcal{O}(\epsilon/[\tau + \log(1/\epsilon)]), \qquad (20)$$

which will determine the gate complexity. In Lemmas 8 and 9 of App. A we also show that it suffices to represent the relevant inputs, i.e., masses and spring constants, with a number of bits that is $r_m = \mathcal{O}(\log(m_{\max}/(m_{\min}\epsilon')))$ and $r_\kappa = \mathcal{O}(\log(1/\epsilon'))$, respectively, to achieve error ϵ' in the block encoding. The gate complexity for the arithmetic is essentially $\mathcal{O}(r_m r_\kappa)$, and there is also another contribution $\mathcal{O}(n)$ to the gate complexity for other operations in the block encoding (e.g, for inequality tests and other state preparations). Hence, we can bound the overall gate complexity as $G = \mathcal{O}(Q \log^2(N m_{\max}/(m_{\min}\epsilon')))$.

Substituting the value of ϵ' from Eq. (20), this gives

$$G = \mathcal{O}\left(Q \log^2\left(\frac{N\tau}{\epsilon} \frac{m_{\max}}{m_{\min}} \right) \right), \qquad (21)$$

which is the expression provided in Thm. 1. To simplify this expression we used the fact that $\log((\tau + \log(1/\epsilon))/\epsilon) \leq \log((\tau + (1/\epsilon))/\epsilon)$. Because we consider complexity for large τ and $1/\epsilon$, we can upper bound that by $\log(\tau/\epsilon^2) = \mathcal{O}(\log(\tau/\epsilon))$.

In Thm. 1 we have not quantified the state preparation complexity (i.e., the complexity of $\mathcal{W}$), since Problem 1 assumes $\mathcal{W}$ is given as an input. However, in many situations this preparation can often be performed efficiently. One example is when we have oracles to separately prepare states with amplitudes proportional to $\vec{x}(0)$ and $\dot{\vec{x}}(0)$. In App. E, we show the query complexity to prepare $|\psi(0)\rangle$ is $Q_{\text{ini}} = \mathcal{O}(\sqrt{E_{\max} d/E})$, where

$$E_{\max} = \frac{m_{\max}}{2}\|\dot{\vec{x}}(0)\|^2 + \frac{\kappa_{\max}}{2}\|\vec{x}(0)\|^2 \qquad (22)$$

is the energy of N uncoupled oscillators of mass $m_{\max}$ and spring constants $\kappa_{\max}$ with similar initial conditions. We also establish the gate complexity $G_{\text{ini}} = \mathcal{O}(Q_{\text{ini}} \log^2(N E_{\max}/(E\epsilon)))$.

V. EXAMPLE APPLICATION

Our quantum algorithm can simulate classical oscillators in time polynomial in n under some assumptions. As we are concerned with dynamics and seek to preserve the speedup, we focus on computing certain time-dependent properties of the system. In particular, we explain how to use the quantum algorithm to solve Problem 2, i.e., to estimate the kinetic energy of all or some oscillators $V \subseteq [N]$ at any time.

Theorem 5. *Let $\mathcal{V}$ be an oracle for V, i.e., a unitary that performs the map $\mathcal{V}|j\rangle = -|j\rangle$ if $j \in V$ and $\mathcal{V}|j\rangle = |j\rangle$ otherwise, and $\delta > 0$. Then Problem 2 can be solved with probability at least $1 - \delta$ by a quantum algorithm that makes $\mathcal{O}(\log(1/\delta)/\epsilon)$ uses of $\mathcal{V}$ and the quantum circuit that prepares $|\psi(t)\rangle$ (along with its inverse and controlled version).*

When $V = [N]$, the potential energy is $U(t) = E - K_V(t)$, and a solution to Problem 2 provides an estimate of $U(t)/E$ as well. As usual, we include controlled applications of the oracle $\mathcal{V}$, which means that the global phase is distinguishable (this way we have that $V = [N]$ is distinct from $V = \{\}$). Although here we focus on estimating kinetic energies,

a similar idea can be used to estimate the potential energies stored on a subset of springs at any time, as we discuss below.

To prove Thm. 5, we note that $K_V(t)/E = \langle\psi(t)|P_V|\psi(t)\rangle$, where $P_V = (\mathbb{1} - \mathcal{V})/2$ is the projector onto V. This expectation, which is the support of $|\psi(t)\rangle$ on the subspace spanned by the basis states whose labels correspond to V (i.e., a subspace of that spanned by the first N basis states), can be obtained by making measurements on many copies of $|\psi(t)\rangle$, but that approach is not optimal (i.e., the scaling is quadratic in $1/\epsilon$). The problem reduces to estimating $\langle\psi(t)|\mathcal{V}|\psi(t)\rangle$ with additive error at most 2ϵ and error probability δ. A method based on high-confidence amplitude estimation [25] then provides the result in Thm. 5.

Hence, when $|\psi(t)\rangle$ can be efficiently prepared and $\mathcal{V}$ can be efficiently implemented, we can obtain an estimate of $K_V(t)/E$ with error ϵ and high probability in time polynomial in n. If in addition E is known or is efficiently computed, this translates to an efficient estimation of $K_V(t)$ with error ϵE.

A related result shows that our quantum algorithm can be applied to estimate the potential energy of a subset of oscillators:

Problem 5. *Given the same inputs as Problem 1 and an oracle for $V \subset [N] \times [N]$, which is now a subset of springs (edges) labeled as (j, k) with $k \geq j$, output an estimate $\hat{u}_V(t) \in \mathbb{R}$ such that*

$$|\hat{u}_V(t) - U_V(t)/E| \leq \epsilon \,, \qquad (23)$$

where

$$U_V(t) := \frac{1}{2} \sum_{j:(j,j)\in V} \kappa_{jj} x_j(t)^2 + $$
$$+ \frac{1}{2} \sum_{k>j:(j,k)\in V} \kappa_{jk}(x_j(t) - x_k(t))^2 \qquad (24)$$

is the potential energy of the springs in V at time t.

By noting that $U_V(t)/E$ is the support of $|\psi(t)\rangle$ on the subspace spanned by the basis states $|j, k\rangle$ such that $(j, k) \in V$, the problem reduces to estimating the expectation of another unitary $\mathcal{V}$ on $|\psi(t)\rangle$. Like in the prior example for the kinetic energy, this can be done efficiently using our quantum algorithm:

Theorem 6. *Let $\mathcal{V}$ be an oracle for V, i.e., a unitary that performs the map $\mathcal{V}|j, k\rangle = -|j, k\rangle$ if $(j, k) \in V$ and $\mathcal{V}|j, k\rangle = |j, k\rangle$ otherwise, and $\delta > 0$. Then, Problem 5 can be solved with probability at least $1 - \delta$ by a quantum algorithm that makes $\mathcal{O}(\log(1/\delta)/\epsilon)$ uses of $\mathcal{W}$ and the quantum circuit that prepares $|\psi(t)\rangle$ (along with its inverse and controlled version).*

VI. IMPOSSIBILITY OF EFFICIENT CLASSICAL SIMULATIONS

An important question is whether our quantum algorithm results in an exponential quantum speedup, or whether it can be efficiently simulated by classical algorithms. We address this question in two ways: (i) by showing that our approach can solve an oracular problem – the "glued trees" problem of Ref. [26] – in poly(n) time, while Thm. 2 is implied by Ref. [26], and (ii) by showing that our approach can simulate any quantum circuit of size L acting on q qubits in poly(L, q) time. More precisely we show that Problem 3 is BQP-complete, thereby proving Thm. 3. Although our results use physical systems that may seem artificial (e.g., resulting interactions between oscillators are not spatially local), they are strong evidence that no polynomial-time (in n) classical algorithm for simulating systems of coupled classical oscillators exists. Our results also add a new problem to the list of "natural" BQP-complete problems [27].

A. Oracle lower bound

At a high level, in the glued trees problem of Ref. [26], we are given oracle access to the adjacency matrix of a sparse graph with an ENTRANCE vertex, and the goal is to find the EXIT vertex. We map this problem to a system of masses and springs by putting a unit mass at each vertex and a spring of unit spring constant at every edge. Then we show that if the system starts at rest except with the ENTRANCE vertex having unit velocity, after polynomial time, the EXIT vertex will have inverse polynomial velocity, and thus it is a special case of Problem 2. Formally, we study the following problem:

Problem 6. *Consider a network of $N = 2^{n+1} - 2$ masses coupled by springs, where $\mathbf{M} = \mathbb{1}$ and $\mathbf{K}$ coincides with the adjacency matrix of a graph consisting of two balanced binary trees of depth n "glued" randomly as in Fig. VI A (i.e., the spring constants are $\kappa_{jk} = 1$ if (j, k) is an edge of the graph or $\kappa_{jk} = 0$ otherwise). Each mass (vertex) of the network is labeled randomly with a bit string of size $2n$. The network contains two special and verifiable masses, ENTRANCE and EXIT, which correspond to the roots of both trees. Given oracle access to $\mathbf{K}$ and the label of the ENTRANCE mass, the problem is to find the label of the EXIT mass.*

This problem is equivalent to that studied in Ref. [26] where the authors presented an efficient
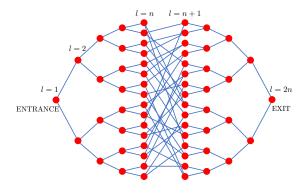
FIG. 2. A network of $N = 2^{n+1} - 2$ coupled oscillators obtained by randomly gluing two binary trees. Each mass is $m_j = 1$ for all $j \in [N]$ and is labeled by a random bit string of size $2n$. Edges denote springs of constant $\kappa_{jk} = 1$. The goal is to find the label of the EXIT mass given the label of the ENTRANCE mass and given oracle access to the network.

quantum walk based algorithm while showing that no classical algorithm can solve the problem efficiently in this oracular setting. In that work the root vertices do not have the extra edge that connects them to a "wall", so they can be easily verified as they are the only vertices of degree two. We add these extra edges to simplify the analysis below; this small change allow us to use some results on the spectral properties of the adjacency matrix already studied in Ref. [26]. With this change, the oracle to access **K** allows us to verify whether a certain vertex is a root or not, since they are the only vertices (or masses) with nonzero diagonal spring constants; i.e., $\kappa_{jj} = 1$ only if j corresponds to ENTRANCE or EXIT and $\kappa_{jj} = 0$ otherwise. Since our problem is a minor modification of the glued trees problem, and any oracle query to our problem can be answered using a single oracle query to the glued trees problem, the original classical lower bound of Ref. [26] implies that our problem requires $2^{\Omega(n)}$ queries to solve classically.

We then show that our quantum algorithm for simulating coupled classical oscillators can also be used to solve Problem 6 efficiently. This is a different approach from that of Ref. [26] as we are not simulating the quantum walk but rather the classical dynamics obtained from Newton's equation with a quantum algorithm. This shows that our quantum algorithm cannot be simulated classically efficiently in the oracular setting in general. Our main claim is the following lemma, which we prove in App. B.

Lemma 7. Let $\vec{x}(0) = (0, 0, \ldots, 0)^T$ and $\dot{\vec{x}}(0) =$

$(1, 0, \ldots, 0)^T$, where the first and last components refer to the initial conditions of the ENTRANCE and EXIT masses in the network of Fig. VIA, respectively. Then, there exists a time $t = \mathcal{O}(\mathrm{poly}(n))$ at which the kinetic energy of the EXIT mass is $K_{\mathrm{EXIT}}(t) := \frac{1}{2}(\dot{x}_{\mathrm{EXIT}}(t))^2 = \Omega(1/\mathrm{poly}(n))$.

Given this lemma, it is easy to see that Problem 6 is a special case of the problem of estimating the kinetic energy, Problem 2. The quantum algorithm can prepare $|\psi(t)\rangle$ efficiently under the conditions of Lemma 7. Because Lemma 7 proves that the kinetic energy of the EXIT mass will be inverse polynomially large after time $t = \mathcal{O}(\mathrm{poly}(n))$, a projective measurement on $|\psi(t)\rangle$ will return the label of the EXIT with probability $K_{\mathrm{EXIT}}(t)/E = \Omega(1/\mathrm{poly}(n))$, where the energy is $E = \frac{1}{2}(\dot{x}_{\mathrm{ENTRANCE}}(0))^2 = \frac{1}{2}$ in this case. Since the EXIT mass can be verified with the oracle by assumption, the probability of finding the label of EXIT can be made a constant close to 1 after $\mathcal{O}(\mathrm{poly}(n))$ repetitions of the previous procedure, giving an overall query and gate complexity $\mathcal{O}(\mathrm{poly}(n)))$. Since Problem 6 requires $2^{\Omega(n)}$ queries to solve classically, this yields the exponential lower bound in Thm. 2.

B. BQP-completeness

To prove this, we start from the standard BQP-complete problem of simulating universal quantum circuits, i.e., the problem of determining some property of the output state $|\phi\rangle = U_L \ldots U_1 |0\rangle^{\otimes q}$, where $L = \mathrm{poly}(q)$. The unitary gates U_l belong to a universal set such as $\{H, X, \mathrm{Toff}\}$, where H and X are single-qubit Hadamard and Pauli X (bit flip) gates, and Toff is the three-qubit Toffoli gate. (Without loss of generality, all Hadamard gates can act on the last qubit, a property that is not necessary but we use it to simplify the presentation of the proof.) The specific problem is that of deciding if $|\phi\rangle$ is essentially the all zero state or has almost no overlap on it; see Problem 7. In App. C we show that this problem reduces to one in which the kinetic energy of a specific oscillator is either exponentially close to zero or at least $1/\mathrm{poly}(q)$ large. This is essentially Problem 3 for $n = \mathcal{O}(q)$.

We use the standard Feynman-Kitaev construction [28, 29] to encode the U_l's into a Hamiltonian with an extra "clock" register. One property of this construction is that evolution under the Hamiltonian is known to apply the sequence of gates $U_L \ldots U_1$ on a given initial state (with $1/\mathrm{poly}(q)$ amplitude). We want this Hamiltonian to be the matrix **A** of a

corresponding system of coupled oscillators. This way we could use the connection between classical and quantum systems discussed in Sec. III, which suggests that the evolved state of the oscillators encodes the amplitudes of $|\phi\rangle$. However, this has two problems:

1. The off-diagonal entries of the Hamiltonian would not be real negative numbers, i.e., they cannot be related to spring constants;

2. The evolution of the oscillators is induced by the $\sqrt{\mathbf{A}}$ rather than $\mathbf{A}$, so the evolution property of the Hamiltonian does not apply.

The first problem is due to the fact that Hadamards have negative matrix entries. To fix this, we observe that the same effect can be obtained using an operator with non-negative entries and an ancilla in the $|-\rangle := \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ state [30, 31]. This operator is no longer unitary, but this poses no additional problems as it is only being encoded in a Hamiltonian that, at this stage, corresponds to a system of oscillators coupled by springs. The second problem is addressed by showing that the spectral properties of $\sqrt{\mathbf{A}}$ are such that there is an appropriate evolution time $t = \text{poly}(q)$ after which the evolved state of the oscillators indeed encodes the output of the quantum circuit with $1/\text{poly}(q)$ amplitude.

Formally, we consider problem instances, i.e., systems of coupled oscillators, where $N = (L + 1)2^{q+1}$, $\mathbf{M} = \mathbb{1}$, and $\mathbf{A} = \mathbf{F}$ is

$$\mathbf{A} = 4\mathbb{1}_N - \sum_{l=1}^{L}(|l\rangle\langle l + 1| + |l + 1\rangle\langle l|) \otimes W_l . \quad (25)$$

The matrix $\mathbb{1}_N$ is the $N \times N$ identity matrix. The $2^{q+1} \times 2^{q+1}$ matrices W_l are real-symmetric and act on one more qubit than do the U_l matrices. We define them as $W_l = U_l \otimes \mathbb{1}_2$ if U_l is the X or Toff gate acting on the space of q qubits. When U_l is a Hadamard gate, which acts on the last qubit (the qth qubit), we replace it by the following 4×4 matrix acting on qubits q and $q + 1$:

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}. \quad (26)$$

This matrix acts like Hadamard on qubit q when qubit $q + 1$ is set to $|-\rangle$. Thus the entries of W_l are non-negative in this case as well, the off-diagonal entries of $\mathbf{A}$ are $\{0, -1/\sqrt{2}, -1\}$, and all diagonal entries are 4. Hence, the off-diagonal entries of the

matrix $\mathbf{K}$ are $\{0, 1/\sqrt{2}, 1\}$ and one can show that $\kappa_{jj} \geq 0$ for all $j \in [N]$. These spring constants can be efficiently accessed using Eq. (25). The gates U_l are 2-sparse, implying that the W_l's are also 2-sparse, and $\mathbf{A}$ and $\mathbf{K}$ are 4-sparse. These properties are outlined in Thm. 3.

For our proof, we consider the initial condition for the N oscillators where $\vec{x}(0) = (0, 0, \ldots, 0)^T$, $\dot{\vec{x}}(0) = (+1, -1, 0, \ldots, 0)^T$, so that the energy is $E = 1$. Labeling each oscillator $j \in [N]$ by (l, r), where $l \in [L + 1]$ and $r \in [2^{q+1}]$, the oscillator under consideration is the one with $l = L + 1$ and $r = 1$. In App. C we show that computing the kinetic energy of this oscillator after time $t = \text{poly}(q) = \text{polylog}(N)$ solves the above BQP-complete problem. In addition, this classical system satisfies all of our conditions for the quantum algorithm to be efficient (i.e., the problem is in BQP).

VII. GENERALIZED COORDINATES

In general, when modeling a classical system using the harmonic approximation, Newton's equation can be simply written as $\ddot{\vec{y}}(t) = -\mathbf{A}\vec{y}(t)$, where $\vec{y}(t) \in \mathbb{R}^N$ encodes the generalized coordinates and $\mathbf{A} \succcurlyeq 0$ is of dimension $N \times N$. To prove Thm. 4 we use a standard approach based on quantum phase estimation [32–34] to first estimate the eigenvalues of $\mathbf{H}^{(2)} := -X \otimes \mathbf{A}$, which are $\gamma_{\eta,j} := (-1)^\eta \lambda_j$, within certain precision that gives the eigenvalues of $\mathbf{H}$ within precision $\mathcal{O}(\epsilon/t)$. Here, $\lambda_j \geq 0$ are the eigenvalues of $\mathbf{A}$, and $\eta \in \{0, 1\}$ determine the eigenvalues of $-X$ as $(-1)^\eta$, with eigenstates $|0_X\rangle = |-\rangle$ and $|1_X\rangle = |+\rangle$. We specifically use a standard QFT-based approach to phase estimation that eschews the need for measurement and in turn maps (within small error) each eigenvector of $\mathbf{H}^{(2)}$ via

$$|\eta_X, \lambda_j\rangle \mapsto |\eta_X, \lambda_j\rangle \left(\sqrt{1 - \delta_{\eta,j}} \sum_{x:|x-\gamma_{\eta,j}| \leq \epsilon_{\text{PE}}} b_x |x\rangle + \right.$$
$$\left. + \sqrt{\delta_{\eta,j}} |\phi_{\eta,j}\rangle \right), \quad (27)$$

where $|\phi_{\eta,j}\rangle$ is some unspecified error state of the ancillas of unit norm, $\sum_x |b_x|^2 = 1$, $\delta_{\eta,j} \leq \delta_{\text{PE}}$, and ϵ_{PE} and δ_{PE} need to be set according to the error requirements. The states $|x\rangle$ encode the eigenvalue estimates of $\mathbf{H}^{(2)}$. From each estimate x, we obtain the eigenvalue estimate of $\mathbf{H}$ by taking its sign and computing a square root, i.e., $\text{sign}(x)\sqrt{|x|}$. Then we

apply a phase factor yielding

$$|\eta_X, \lambda_j\rangle \left(\sqrt{1 - \delta_{\eta,j}} \sum_{x:|x-\gamma_{\eta,j}|\leq\epsilon_{\mathrm{PE}}} e^{-it\,\mathrm{sign}(x)\sqrt{|x|}} b_x \,|x\rangle \right.$$
$$\left. + \sqrt{\delta_{\eta,j}}\,|\phi'_{\eta,j}\rangle \right) . \qquad (28)$$

Last, we invert quantum phase estimation and other operations to (approximately) uncompute the estimated eigenvalues. The result is, for a value of the error tolerances where $\delta_{\mathrm{PE}} = \mathcal{O}(\epsilon^2)$ and

$$\epsilon_{\mathrm{PE}} = \mathcal{O}\left(\max\left(\frac{\epsilon}{t\sqrt{\|\mathbf{A}^{-1}\|}}, \frac{\epsilon^2}{t^2}\right)\right), \qquad (29)$$

a unitary approximating $e^{-it\mathbf{H}}$ within error ϵ. The result is implied by the complexity of quantum phase estimation that requires $\mathcal{O}(\|\mathbf{A}\|_{\max} d \log(1/\delta_{\mathrm{PE}})/\epsilon_{\mathrm{PE}})$ queries to the oracle. See App. D for details.

VIII. CONCLUSION

We introduced an approach to simulating systems of coupled classical oscillators using a quantum computer. We showed how to map Newton's equations for these dynamics (coupled second order ordinary differential equations) to the Schrödinger equation (a first order partial differential equation), with polylogarithmic overhead in the number of oscillators under certain assumptions. Using this mapping, we demonstrated that any classical algorithm that simulates these classical dynamics requires at least $2^{\Omega(n)}$ queries to the oracle, and that they can provide solutions to BQP-complete problems, and would thus require exponential time to solve classically under reasonable complexity theoretic assumptions. We also generalized our approach to the simulation of other classical harmonic systems.

While providing a large quantum advantage in certain contexts, these techniques also have significant limitations. For example, the approach is only efficient for computing particularly large or global properties and when masses and spring constants can be computed in time polylogarithmic in system size. Another feature of our algorithm is that its complexity is (almost) linear in the evolution time t, a scaling that might not be avoided in general [19, 35, 36], being efficient only if t is also polylogarithmic in system size. This would discourage applications where, for example, $N = \mathrm{poly}(t)$ (and when fast forwarding is

not possible). This feature is expected to arise when simulating physical systems with geometrically-local interactions, and for initial states that are locally supported. In these examples, the relevant system size is determined by the "light cone", whose size in D spatial dimensions would scale as $N \sim t^D$. Nevertheless, even for these examples our quantum algorithm would still result in significant quantum speedups (e.g., super-quadratic), suggesting a new application area for quantum computers [37].

As many systems from molecular vibrations, to structural mechanics, to electrical grids, to neuronal activation can be modelled within the harmonic approximation, and since interesting dynamical features can appear at relatively short times (e.g., t independent of N), we expect that there exist specific applications that meet all the requirements for our quantum algorithm to be efficient. Such applications will then benefit from this speedup with appreciable real-world impact; identifying them is one important next step in this line of research.

Last, we note that our classical-to-quantum reduction provides yet another way to think about quantum algorithms. For example, once mapped to a one dimensional system, the glued-trees problem of masses and springs becomes a simple wave propagation problem. Since waves propagate ballistically, it is now clear why our quantum algorithm, whose complexity is mainly dominated by evolution time but not by the number of masses, works in this case. Another example results from Grover's classical system of coupled pendulums [18] that, after using our reduction, provides another way to solve the unstructured search problem with $\mathcal{O}(t)$ queries, where $t = \mathcal{O}(\sqrt{N})$.

[1] Richard P. Feynman, "Simulating physics with computers," International Journal of Theoretical Physics **21**, 467–488 (1982).

[2] Seth Lloyd, "Universal Quantum Simulators," Science **273**, 1073–1078 (1996).

[3] Leonard M. Adleman, Jonathan DeMarrais, and Ming-Deh A. Huang, "Quantum Computability," SIAM Journal on Computing **26**, 1524–1540 (1997).

[4] P W Shor, "Algorithms for quantum computation: discrete logarithms and factoring," Proceedings 35th Annual Symposium on Foundations of Computer Science , 124–134 (1994).

[5] Dominic W. Berry, "High-order quantum algorithm for solving linear differential equations," Journal of Physics A: Mathematical and Theoretical **47**, 105301 (2014).

[6] B. D. Clader, B. C. Jacobs, and C. R. Sprouse, "Preconditioned quantum linear system algorithm," Physical Review Letters **110**, 250504 (2013).

[7] B. D. Clader, B. C. Jacobs, and C. R. Sprouse, "Publisher's Note: Preconditioned Quantum Linear System Algorithm [Phys. Rev. Lett. 110, 250504 (2013)]," Physical Review Letters **111**, 49903 (2013).

[8] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd, "Quantum Algorithm for Linear Systems of Equations," Physical Review Letters **103**, 150502 (2009).

[9] Andrew M. Childs, Robin Kothari, and Rolando D. Somma, "Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision," SIAM Journal on Computing **46**, 1920–1950 (2017).

[10] Yiğit Subaşı, Rolando D. Somma, and Davide Orsucci, "Quantum Algorithms for Systems of Linear Equations Inspired by Adiabatic Quantum Computing," Physical Review Letters **122**, 060504 (2019).

[11] Pedro C. S. Costa, Stephen Jordan, and Aaron Ostrander, "Quantum algorithm for simulating the wave equation," Physical Review A **99**, 012323 (2019).

[12] Shi Jin, Nana Liu, and Yue Yu, "Quantum simulation of partial differential equations via Schrodingerisation," arXiv:2212.13969 (2022).

[13] Dong An, Jin-Peng Liu, Daochen Wang, and Qi Zhao, "A theory of quantum differential equation solvers: limitations and fast-forwarding," arXiv:2211.05246 (2022).

[14] Lev D. Landau and Evgeny M. Lifshitz, *Mechanics*, 3rd ed., Vol. 1 (Butterworth-Heinemann, 2010).

[15] John David Jackson, *Classical Electrodynamics*, 3rd ed., Vol. 1 (Wiley, 1998).

[16] Hartmut Neven and Ad Aertsen, "Rate coherence and event coherence in the visual cortex: a neuronal model of object recognition," Biological Cybernetics **67**, 309–322 (1992).

[17] E. Bright Wilson, J. C. Decius, and P. C. Cross, *Molecular Vibrations: The Theory of Infrared and Raman Vibrational Spectra* (Courier Corporation, New York, 1980).

[18] Lov K. Grover and Anirvan M. Sengupta, "Classical analog of quantum search," Physical Review A **65**, 032319 (2002).

[19] Dominic W. Berry, Graeme Ahokas, Richard Cleve, and Barry C. Sanders, "Efficient quantum algorithms for simulating sparse Hamiltonians," Comm. Math. Phys. **270**, 359 (2007).

[20] Dominic W. Berry and Andrew M. Childs, "Black-box Hamiltonian simulation and unitary implementation," Quantum Information and Computation **12**, 29–62 (2012), arXiv:0910.4157.

[21] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma, "Simulating Hamiltonian dynamics with a truncated Taylor series," Physical Review Letters **114**, 090502 (2015).

[22] Dominic W. Berry, Andrew M. Childs, and Robin Kothari, "Hamiltonian simulation with nearly optimal dependence on all parameters," in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science* (2015) pp. 792–809.

[23] Guang Hao Low and Isaac L. Chuang, "Optimal Hamiltonian simulation by quantum signal processing," Physical Review Letters **118**, 010501 (2017).

[24] Guang Hao Low and Isaac L. Chuang, "Hamiltonian simulation by qubitization," Quantum **3**, 163 (2019).

[25] Emanuel Knill, Gerardo Ortiz, and Rolando D. Somma, "Optimal quantum measurements of expectation values of observables," Physical Review A **75**, 012328 (2007).

[26] Andrew M Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A Spielman, "Exponential algorithmic speedup by a quantum walk," in *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing* (2003) pp. 59–68.

[27] Pawel Wocjan and Shengyu Zhang, "Several natural BQP-complete problems," arXiv:quant-ph/0606179 (2006).

[28] Richard P. Feynman, "Quantum mechanical computers," Foundations of Physics **16**, 507–531 (1986).

[29] A. Yu. Kitaev, A. H. Shen, and M. N. Vyalyi, *Graduate Studies in Mathematics*, Vol. 47 (Amer Mathematical Society, 2002) p. 257.

[30] Stephen P. Jordan, David Gosset, and Peter J. Love, "Quantum-Merlin-Arthur–complete problems for stoquastic Hamiltonians and Markov matrices," Physical Review A **81**, 032331 (2010).

[31] Andrew M. Childs, David Gosset, and Zak Webb, "The Bose-Hubbard model is QMA-complete," Theory of Computing **11**, 491–603 (2015).

[32] A. Kitaev, "Quantum measurements and the Abelian stabilizer problem," arXiv:quant-ph/9511026 (1995).

[33] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, "Quantum algorithms revisited," Proc. R. Soc. Lond. A **454**, 339–354 (1998).

[34] Yuval R. Sanders, Dominic W. Berry, Pedro C. S. Costa, Louis W. Tessler, Nathan Wiebe, Craig Gidney, Hartmut Neven, and Ryan Babbush, "Compilation of Fault-Tolerant Quantum Heuristics for Combinatorial Optimization," PRX Quantum **1**, 020312–020382 (2020).

[35] Y. Atia and D. Aharonov, "Fast-forwarding of Hamiltonians and exponentially precise measurements," Nat. Comm. **8**, 1572 (2017).

[36] Jeongwan Haah, Matthew B. Hastings, Robin Kothari, and Guang Hao Low, "Quantum algorithm for simulating real time evolution of lattice Hamiltonians," in Proceedings of the IEEE 59th Symposium on Foundations of Computer Science (2018) p. 350.

[37] Ryan Babbush, Jarrod R. McClean, Michael Newman, Craig Gidney, Sergio Boixo, and Hartmut Neven, "Focus beyond Quadratic Speedups for Error-Corrected Quantum Advantage," PRX Quantum **2**, 010103 (2021).

[38] Yuval R. Sanders, Guang Hao Low, Artur Scherer, and Dominic W. Berry, "Black-box quantum state preparation without arithmetic," Physical Review Letters **122**, 020502 (2019).

[39] Yaoyun Shi, "Both Toffoli and controlled-NOT need little help to do universal quantum computation," arXiv:quant-ph/0205115 (2002).

[40] Dorit Aharonov, "A simple proof that Toffoli and Hadamard are quantum universal," arXiv:quant-ph/0301040 (2003).

[41] Sergio Boixo, Emanuel Knill, and Rolando Somma, "Eigenpath traversal by phase randomization," Quantum Info. Comput. **9**, 0833–0855 (2009).

[42] Ruggero Vaia and Lidia Spadini, "Persymmetric Jacobi matrices with square-integer eigenvalues and dispersionless mass-spring chains," Linear Algebra and its Applications **585**, 164–177 (2020).

[43] Hugo Schérer, Luc Vinet, and Alexei Zhedanov, "Analytic "Newton's cradles" with perfect transfer and fractional revival," Annals of Physics **439**, 168790 (2022).

[44] Jean-Michel Lemay, Luc Vinet, and Alexei Zhedanov, "The para-Racah polynomials," Journal of Mathematical Analysis and Applications **438**, 565–577 (2016).

[45] Dominic W. Berry, Yuan Su, Casper Gyurik, Robbie King, Joao Basso, Alexander Del Toro Barba, Abhishek Rajput, Nathan Wiebe, Vedran Dunjko, and Ryan Babbush, "Quantifying quantum advantage in topological data analysis," arXiv:2209.13581 (2022).

[46] Dominic W. Berry, Mária Kieferová, Artur Scherer, Yuval R. Sanders, Guang Hao Low, Nathan Wiebe, Craig Gidney, and Ryan Babbush, "Improved techniques for preparing eigenstates of fermionic Hamiltonians," npj Quantum Information **4**, 22 (2018).

[47] Mikko Möttönen, Juha J. Vartiainen, Ville Bergholm, and Martti M. Salomaa, "Transformation of quantum states using uniformly controlled rotations," Quantum Info. Comput. **5**, 467–473 (2005).

[48] Theodore J. Yoder, Guang Hao Low, and Isaac L. Chuang, "Fixed-point quantum search with an optimal number of queries," Physical Review Letters **113**, 210501 (2014).

[49] Lov K. Grover, "A fast quantum mechanical algorithm for database search," in Proceedings of the 28th ACM Symposium on Theory of Computing (1996) pp. 212–219.

[50] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani, "Strengths and weaknesses of quantum computing," SIAM Journal on Computing **26**, 1510–1523 (1997).

[51] Andrew M Childs, Jin-Peng Liu, and Aaron Ostrander, "High-precision quantum algorithms for partial differential equations," Quantum **5**, 574 (2021).

[52] Hari Krovi, "Improved quantum algorithms for linear and nonlinear differential equations," Quantum **7**, 913 (2023).

[53] Andrew M. Childs, Robin Kothari, and Rolando D. Somma, "Quantum linear systems algorithm with exponentially improved dependence on precision," SIAM J. Comp. **46**, 1920 (2017).

[54] Pedro C.S. Costa, Dong An, Yuval R. Sanders, Yuan Su, Ryan Babbush, and Dominic W. Berry, "Optimal scaling quantum linear-systems solver via discrete adiabatic theorem," PRX Quantum **3**, 040303 (2022).

[55] Anirban N. Chowdhury and Rolando D. Somma, "Quantum algorithms for Gibbs sampling and hitting-time estimation," Quant. Inf. Comp. **17**, 0041 (2017).

APPENDIX

Appendix A: Access models and block encoding of B and H

We provide more details on the access model used for our quantum algorithm, which is similar to that used in prior quantum algorithms [19, 20]. The black box or oracle $\mathcal{S}$ allows us to perform the map

$$|j, \ell\rangle \to |j, a(j, \ell)\rangle \ , \tag{A1}$$

for any $j \in [N] := \{1, \ldots, N\}$ and $\ell \in [d]$, where d is the maximum number of nonzero entries in any row of $\mathbf{K}$ (i.e., the sparsity), and $a(j, \ell)$ is the column index of the ℓth nonzero entry in the jth row of $\mathbf{K}$. The same black box allows us to perform the maps $|j, k, z\rangle \to |j, k, z \oplus \bar{\kappa}_{jk}\rangle$ and $|j, z\rangle \to |j, z \oplus \bar{m}_j\rangle$, for any $j, k \in [N]$, where z, $\bar{\kappa}_{jk}$, and $\bar{m}_j$ are assumed to be given as bit strings. (One could use three different oracles for these maps, but we combine them into one that we call $\mathcal{S}$.) Specifically, if $m_{\max} \geq m_j$ for all $j \in [N]$, then $|\bar{m}_j\rangle$ denotes a basis state determined from the bits in the binary fraction $m_j = m_{\max}[.b_{j,1}b_{j,2}\ldots]$, where $b_{j,i} \in \{0, 1\}$. Similarly, if $\kappa_{\max} \geq \kappa_{jk}$ for all $j, k \in [N]$, then $|\bar{\kappa}_{jk}\rangle$ denotes a basis state determined from the bits in the binary fraction $\kappa_{jk} = \kappa_{\max}[.c_{jk,1}c_{jk,2}\ldots]$, where $c_{jk,i} \in \{0, 1\}$. This access model simplifies some calculations and does not require knowing the important quantities in any specific units.

Similar to Refs. [19, 20], we assume that any number of bits can be given as the output of $\mathcal{S}$, with it still being accounted for as a single oracle call. However, for computations with this output, we will only use a limited number of bits, the choice of which is governed by the parameters of the problem, particularly the approximation errors. In particular, we denote by r_m and r_κ the number of bits used to represent m_j and κ_{jk}. Then $m_j = m_{\max}\bar{m}_j/2^{r_m}$ and $\kappa_{jk} = \kappa_{\max}\bar{\kappa}_{jk}/2^{r_\kappa}$. These approximations as well as the choices for r_m and r_κ are discussed below, where we show how to use $\mathcal{S}$ to gain access to $\mathbf{B}$ or $\mathbf{H}$.

1. Access to B and H

We explain how to use $\mathcal{S}$ and other gates to access $\mathbf{H}$, the Hamiltonian in Eq. (12). This access is required by Hamiltonian simulation methods in Refs. [21–24]. We do this by showing a block encoding of the relevant matrices using known methods for state preparation via inequality testing [38], i.e., by constructing a unitary such that one of its blocks approximates $\mathbf{H}/\Lambda$, where Λ is needed for normalization reasons and given below. We explain how to implement that unitary operation using elementary gates.

In our construction, we start by providing a block encoding of $\mathbf{B}$ (and $\mathbf{B}^\dagger$). This is the $N \times M$ matrix discussed in Sec. III. Because $M = N(N+1)/2$ is not a power of two in general ($N = 2^n$), it is convenient to pad $\mathbf{B}$ with zeroes so that its dimension is $N \times N^2$, instead. That is, here we consider and describe the simulation of $\mathbf{H}$ in a larger-dimensional space, where many amplitudes of the evolved state will be zero and the others will coincide with those of $|\psi(t)\rangle$ in Eq. (2). With this padding, the matrices $\mathbf{B}$ and $\mathbf{H}$ will be then of dimension $N \times N^2$ and $2N^2 \times 2N^2$, respectively, in the following analyses. First we prove:

Lemma 8. *Let $\epsilon' > 0$, $m_{\max} \geq m_j \geq m_{\min}$ and $\kappa_{\max} \geq \kappa_{jk}$ for all $j, k \in [N]$, and $\aleph := \kappa_{\max}/m_{\min}$. Consider the d-sparse $N \times N^2$ matrix $\mathbf{B}$ obtained from Sec. III (with the padding). Then, there exists a unitary $\mathcal{U}_{\mathbf{B}}$ acting on $2n + r + 2$ qubits with $r = \mathcal{O}(\log(1/\epsilon'))$ that provides a block encoding of $\mathbf{B}$ to within additive error ϵ' as follows:*

$$\left\| (\mathbb{1}_N \otimes \langle 0|^{\otimes n} \otimes \langle 0|^{\otimes r+2}) \mathcal{U}_{\mathbf{B}} (\mathbb{1}_N \otimes \mathbb{1}_N \otimes |0\rangle^{\otimes r+2}) - \frac{1}{\sqrt{2\aleph d}}\mathbf{B} \right\| \leq \epsilon' \ . \tag{A2}$$

The quantum circuit that implements $\mathcal{U}_{\mathbf{B}}$ makes $\mathcal{O}(1)$ uses of $\mathcal{S}$ and its inverse, in addition to $\mathcal{O}(n + \log^2(m_{\max}/(m_{\min}\epsilon')))$ two-qubit gates.

Next, we use this result to provide a block encoding for the $2N^2 \times 2N^2$ matrix $\mathbf{H}$:

Lemma 9. *Let $\epsilon' > 0$, $m_j \geq m_{\min}$ and $\kappa_{\max} \geq \kappa_{jk}$ for all $j, k \in [N]$, and $\aleph := \kappa_{\max}/m_{\min}$. Consider the d-sparse $2N^2 \times 2N^2$ matrix $\mathbf{H}$ obtained from Sec. III (with the padding). Then, there exists a unitary $\mathcal{U}_{\mathbf{H}}$*

acting on $2n + r + 4$ qubits with $r = \mathcal{O}(\log(1/\epsilon'))$ that provides a block encoding of $\mathbf{H}$ to within additive error ϵ' as follows:

$$\left\| \langle 0|^{\otimes r+3} \, \mathcal{U}_{\mathbf{H}} \, |0\rangle^{\otimes r+3} - \frac{1}{\sqrt{2\aleph d}} \mathbf{H} \right\| \leq \epsilon' \, . \tag{A3}$$

The quantum circuit that implements $\mathcal{U}_{\mathbf{H}}$ uses a controlled version of $\mathcal{U}_{\mathbf{B}}$ and its inverse once, and additional $\mathcal{O}(n)$ two-qubit gates (e.g., CNOTs and other single qubit gates).

Combining Lem. 8 and Lem. 9, a block encoding for $\mathbf{H}/\Lambda$, where $\Lambda := \sqrt{2\aleph d}$, can be implemented within additive error $\mathcal{O}(\epsilon')$ in spectral norm using the oracles $\mathcal{S}$, together with its inverse and controlled versions, $\mathcal{O}(1)$ times, in addition to $\mathcal{O}(n + \log^2(m_{\max}/(m_{\min}\epsilon')))$ two-qubit gates.

2. State preparation using inequality testing

Before presenting the proofs of the lemmas, we revisit how inequality testing can be used for state preparation [38], since our constructions use this approach, which is more efficient than controlled rotation of an ancilla qubit as in Ref. [8]. Let $\{\beta_1, \ldots, \beta_N\}$ be such that $\beta_j \geq 0$ can be expressed using r bits (see below) and $\beta \geq \beta_j$ for all $j \in [N]$. Assume we have access to an oracle that, on input $|j, z\rangle$ outputs $|j, z \oplus \bar{\beta}_j\rangle$, where z and $\bar{\beta}_j \in [2^r]$ are given as bit strings of size r (we will fix $z = 0 \ldots 0$). That is, in this notation, $|\bar{\beta}_j\rangle$ is a basis state, and the bits are obtained from the binary fraction $\beta_j = \beta[.b_1 \ldots b_r]$. The method in Ref. [38] can be used to perform:

$$|j\rangle \, |0\rangle \to |j\rangle \left(\frac{\beta_j}{\beta} |0\rangle^{\otimes r} |0\rangle + |\omega_j\rangle \right) \, , \tag{A4}$$

where $|\omega_j\rangle$ is a state orthogonal to $|0\rangle^{\otimes r} |0\rangle$. The basic steps of the method are as follows:

1. Apply the oracle to compute $\bar{\beta}_j$, i.e., perform the map $|j\rangle \mapsto |j, \bar{\beta}_j\rangle$.

2. Prepare the equal superposition state of r ancilla qubits, i.e., $\frac{1}{2^{r/2}} \sum_{x=1}^{2^r} |x\rangle$.

3. Apply inequality testing to prepare $|j, \bar{\beta}_j\rangle \frac{1}{2^{r/2}} \left(\sum_{x=1}^{\bar{\beta}_j} |x\rangle |0\rangle + \sum_{\bar{\beta}_j+1}^{2^r} |x\rangle |1\rangle \right)$.

4. Apply Hadamards on the r ancilla qubits.

5. Apply the inverse of the oracle to reverse the computation of $\bar{\beta}_j$, i.e., perform the map $|j, \bar{\beta}_j\rangle \mapsto |j, 0\rangle$.

After these operations there will be amplitude $\bar{\beta}_j/2^r = \beta_j/\beta$ on $|0\rangle^{\otimes r} |0\rangle$. This is because the amplitude can be found by taking the inner product of the state in step 3 with $\frac{1}{2^{r/2}} \sum_{x=1}^{2^r} |x\rangle |0\rangle$. The method requires one use of the oracle and its inverse, in addition to $\mathcal{O}(r)$ Hadamard gates for steps 1 and 4, and $\mathcal{O}(r)$ gates for the inequality test in step 3. It is also possible to compute more complicated functions of the oracle by rearranging the inequality in step 3, which is the method we will use here.

3. Proof of Lemma 8

We consider a block encoding of $\mathbf{B}^\dagger$ for simplicity, and later use it to provide a block encoding of $\mathbf{B}$ by taking the conjugate transpose. Using the padding described above, the dimension of $\mathbf{B}^\dagger$ is $N^2 \times N$ and its entries are of the form $\sqrt{\kappa_{jk}/m_j}$ or zero; specifically, the definition of $\mathbf{B}$ in Sec. III (obtained from Eq. (16)) implies

$$\mathbf{B}^\dagger |j\rangle = \left(\sum_{k \geq j} \sqrt{\frac{\kappa_{jk}}{m_j}} |j\rangle |k\rangle \right) - \left(\sum_{k < j} \sqrt{\frac{\kappa_{jk}}{m_j}} |k\rangle |j\rangle \right) \, . \tag{A5}$$

14

We will construct a unitary $\mathcal{U}_{\mathbf{B}}^{\dagger}$ that is a block encoding of $\mathbf{B}^{\dagger}$ following simple steps and then explain how these can be simulated with quantum circuits. The steps require using a work register of qubits for some computations that we discard at the end.

1. Apply a unitary that performs the map

$$|0\rangle^{\otimes n} \mapsto \frac{1}{\sqrt{d}} \sum_{\ell=1}^{d} |\ell\rangle \ . \tag{A6}$$

2. Apply the oracle $\mathcal{S}$ for the positions of non-zero entries of $\mathbf{K}$ to map $|\ell\rangle$ to $|a(j,\ell)\rangle$ according to Eq. (A1).

3. Apply the oracle $\mathcal{S}$ two more times to compute $\bar{m}_j$ and $\bar{\kappa}_{jk}$, i.e., perform the map $|j,0\rangle \mapsto |j,\bar{m}_j\rangle$ and $|j,k,0\rangle \mapsto |j,k,\bar{\kappa}_{jk}\rangle$.

4. Prepare the equal superposition state of r ancilla qubits, i.e., $\frac{1}{2^{r/2}} \sum_{x=1}^{2^r} |x\rangle$, where r is given below.

5. Using coherent arithmetic, compute the square of x and multiplications needed for the inequality test

$$\frac{\kappa_{\max}\bar{\kappa}_{jk}}{2^{r_\kappa}} \leq \frac{x^2}{2^{2r}} \aleph \frac{m_{\max}\bar{m}_j}{2^{r_m}} , \tag{A7}$$

where r_κ and r_m are the numbers of bits of $\bar{\kappa}_{jk}$ and $\bar{m}_j$, respectively, also determined below. This gives the factor in the amplitude approximately $\sqrt{\kappa_{jk}/(m_j\aleph)}$.

6. Reverse the computation of the arithmetic for Eq. (A7) and the oracles for $\bar{m}_j$ and $\bar{\kappa}_{jk}$. This transforms the working registers back to an all-zero state.

7. Apply inequality testing (outputting the result in an ancilla qubit) and a controlled-swap operation to perform the map $|j\rangle |k\rangle |0\rangle \mapsto |k\rangle |j\rangle |1\rangle$ if $k < j$ or leave the state $|j\rangle |k\rangle |0\rangle$ invariant otherwise.

8. Apply HZ on the ancilla qubit of the previous step to implement $|0\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|1\rangle \mapsto \frac{1}{\sqrt{2}}(-|0\rangle + |1\rangle)$.

The previous sequence of unitaries define $\mathcal{U}_{\mathbf{B}}^{\dagger}$. To show that the method is correct, consider steps 1-6. If we discard the working registers (used to store $\bar{m}_j, \bar{\kappa}_{jk}$ and perform the arithmetic), these steps combined implement approximately

$$|j\rangle |0\rangle^{\otimes n+r} |0\rangle \mapsto |j\rangle \frac{1}{\sqrt{\aleph d}} \sum_k |k\rangle \sqrt{\frac{\kappa_{jk}}{m_j}} |0\rangle^{\otimes r} |0\rangle + |\omega_j\rangle \ , \tag{A8}$$

for some state $|\omega_j\rangle$ that is orthogonal to $|0\rangle^{\otimes r} |0\rangle$ on the ancilla qubits. We want the the error in this approximation to be $\mathcal{O}(\epsilon')$, and we show how to achieve this below. Step 7 is then needed to rearrange the sum depending on whether $k < j$ or $k \geq j$, according to Eq. (A5). This step transforms Eq. (A8) to

$$\frac{1}{\sqrt{\aleph d}} \sum_{k \geq j} |j\rangle |k\rangle |0\rangle \sqrt{\frac{\kappa_{jk}}{m_j}} |0\rangle^{\otimes r} |0\rangle + \frac{1}{\sqrt{\aleph d}} \sum_{k < j} |k\rangle |j\rangle |1\rangle \sqrt{\frac{\kappa_{jk}}{m_j}} |0\rangle^{\otimes r} |0\rangle + |\omega_j'\rangle \ , \tag{A9}$$

where $|\omega_j'\rangle$ is still orthogonal to $|0\rangle^{\otimes r} |0\rangle$ on the ancilla qubits. Applying step 8 to Eq. (A9), and considering the part of the state where the ancillas are in $|0\rangle |0\rangle^{\otimes r} |0\rangle$ only, we obtain

$$\frac{1}{\sqrt{2\aleph d}} \left(\sum_{k \geq j} \sqrt{\frac{\kappa_{jk}}{m_j}} |j\rangle |k\rangle - \sum_{k < j} \sqrt{\frac{\kappa_{jk}}{m_j}} |k\rangle |j\rangle \right) |0\rangle |0\rangle^{\otimes r} |0\rangle \ . \tag{A10}$$

This coincides with Eq. (A5) if we drop the normalization factor and project onto the subspace specified by $|0\rangle |0\rangle^{\otimes r} |0\rangle$ of the ancillas. Hence, the sequence of steps defines the desired unitary $\mathcal{U}_{\mathbf{B}}^{\dagger}$ that satisfies, for all $j \in [N]$,

$$\langle 0|^{\otimes r+2} \mathcal{U}_{\mathbf{B}}^{\dagger} |j\rangle |0\rangle^{\otimes n} |0\rangle^{\otimes r+2} \approx_{\epsilon'} \frac{1}{\sqrt{2\aleph d}} \mathbf{B}^{\dagger} |j\rangle \ . \tag{A11}$$

Equivalently, $(\mathbb{1}_N \otimes \langle 0|^{\otimes n} \otimes \langle 0|^{\otimes r+2}) \mathcal{U}_{\mathbf{B}} (\mathbb{1}_N \otimes \mathbb{1}_N \otimes |0\rangle^{\otimes r+2}) \approx_{\epsilon'} \frac{1}{\sqrt{2\aleph d}} \mathbf{B}$.

Note that the factor of $1/\sqrt{d}$ comes from a single sparse state preparation (with amplitudes $\sqrt{\kappa_{jk}/m_j}$). In contrast, the block encoding based on [20] involves a matched preparation and inverse preparation, each of which gives a factor of $1/\sqrt{d}$ for an overall factor of $1/d$ for Hamiltonian simulation. In particular, the state preparation as in Lemma 4 of [20] gives a factor of $1/\sqrt{d}$, then the step of the walk as in Eq. (23) of [20] involves a matched preparation and inverse preparation to implement the reflection.

The above steps can be implemented with a quantum circuit as follows. The unitary in step 1 (Eq. (A6)) is simply obtained from the action of $\log(d)$ Hadamard gates if d is a power of 2, or otherwise can be performed with high precision by amplitude amplification (see App. E.2 of [34]). The gate complexity is $\mathcal{O}(\log d) = \mathcal{O}(n)$, with an amplitude for success that is very close to 1. When allowing arbitrary qubit rotations in the gate set, as we do here, then the amplitude for success can be made exactly one, so we need no correction here. The gate complexity in step 4 depends on the number of bits needed to represent m_j, κ_{jk}, and x with sufficient accuracy. The leading-order gate complexity is $\mathcal{O}(r^2 + r r_m)$ for computing x^2 and $x^2 \times \bar{m}_j$. We also need to multiply by $\aleph m_{\max}/\kappa_{\max} = m_{\max}/m_{\min}$. The number of bits needed is no more than that in $x^2 \times \bar{m}_j$, which is $\mathcal{O}(r + r_m)$, giving gate complexity $\mathcal{O}((r + r_m)^2)$. The divisions by powers of 2 just involve bit shifts with no gate cost. The gate complexity for the inequality test is linear in the number of bits, so is smaller than the multiplication cost.

It suffices to set $r = \mathcal{O}(\log(1/\epsilon'))$ for overall precision $\mathcal{O}(\epsilon')$, since this choice would imply that the coefficients in Eq. (A8) are given with that precision. To choose r_m, r_κ, we note that our computations of m_j and κ_{jk} are effectively done within additive error $\delta_m = \mathcal{O}(m_{\max}/2^{r_m})$ and $\delta_\kappa = \mathcal{O}(\kappa_{\max}/2^{r_\kappa})$, respectively, since $\bar{m}_j$ and $\bar{\kappa}_{jk}$ give m_j and κ_{jk} relative to $m_{\max}$ and $\kappa_{\max}$. That is, we are effectively giving the factor in the amplitude

$$\sqrt{\frac{\kappa_{jk} + \delta_\kappa}{(m_j + \delta_m)\aleph}} \ . \tag{A12}$$

For additive error $\mathcal{O}(\epsilon')$ in the above, we can let $\delta_m = \mathcal{O}(m_j \epsilon')$, i.e., m_j is computed within multiplicative error $\mathcal{O}(\epsilon')$. This choice would imply $r_m = \mathcal{O}(\log(m_{\max}/(m_{\min}\epsilon')))$. Similarly using the error propagation formula for κ_{jk} indicates that the error is largest for small κ_{jk}, so the choice of r_κ would depend on $\kappa_{\min}$. However, the worst the error can be is rounding down a value by δ_κ to zero, which would imply an error $\mathcal{O}(\sqrt{\delta_\kappa/\kappa_{\max}})$. That implies we can choose $\delta_\kappa = \mathcal{O}(\kappa_{\max}(\epsilon')^2)$ and therefore $r_\kappa = \mathcal{O}(\log(1/\epsilon'))$.

These choices of numbers of bits imply the complexity of the squaring and multiplications at most $\mathcal{O}(\log^2(m_{\max}/(m_{\min}\epsilon')))$. Last, all unitaries in steps 7 and 8 can be implemented with gate complexity $\mathcal{O}(n)$ using standard techniques. These imply the gate complexities stated in Lem. 8.

$\square$

4. Proof of Lemma 9

The result of Lem. 9 is a direct consequence of Lem. 8. Recall that

$$\mathbf{H} = -\begin{pmatrix} \mathbf{0} & \mathbf{B} \\ \mathbf{B}^{\dagger} & \mathbf{0} \end{pmatrix} \ , \tag{A13}$$

where we use $\mathbf{0}$ to denote all-zero matrices whose dimensions are clear from context. Assuming that $\mathbf{B}$ is of dimension $N \times N^2$ following Lem. 8, then $\mathbf{H}$ would be of dimension $(N + N^2) \times (N + N^2)$. However, because

it is easier to work with square matrices, we will further pad $\mathbf{B}$ and $\mathbf{B}^\dagger$ with more zeroes, to make them of dimension $N^2 \times N^2$, implying that the dimension of $\mathbf{H}$ is now $(2N^2) \times (2N^2)$. That is, $\mathbf{H}$ acts on a space of $2n+1$ qubits. (The evolved state will have nonzero amplitude on a subspace of dimension $N+M$ only, where $M = N(N+1)/2$.) The previous padding is equivalent to replacing $\mathbf{B}\,|j,k\rangle \to (\mathbf{B}\,|j,k\rangle) \otimes |0\rangle^{\otimes n}$ and $\langle j,k|\mathbf{B}^\dagger \to (\langle j,k|\mathbf{B}^\dagger) \otimes \langle 0|^{\otimes n}$.

With this small modification, Lem. 8 implies

$$\frac{\mathbf{H}}{\sqrt{2\aleph d}} =$$
$$-\begin{pmatrix} \mathbf{0} & (\mathbb{1}_N \otimes |0\rangle\langle 0|^{\otimes n} \otimes \langle 0|^{\otimes r+2})\mathcal{U}_\mathbf{B}(\mathbb{1}_N \otimes \mathbb{1}_N \otimes |0\rangle^{\otimes r+2}) \\ (\mathbb{1}_N \otimes \mathbb{1}_N \otimes \langle 0|^{\otimes r+2})\mathcal{U}_\mathbf{B}^\dagger(\mathbb{1}_N \otimes |0\rangle\langle 0|^{\otimes n} \otimes |0\rangle^{\otimes r+2}) & \mathbf{0} \end{pmatrix},$$
$$\tag{A14}$$

or, equivalently, $\frac{\mathbf{H}}{\sqrt{2\aleph d}} = \langle 0|^{\otimes r+2} \tilde{\mathcal{U}}_\mathbf{H} |0\rangle^{\otimes r+2}$, where

$$\tilde{\mathcal{U}}_\mathbf{H} := -\,|0\rangle\langle 1| \otimes ((\mathbb{1}_N \otimes |0\rangle\langle 0|^{\otimes n} \otimes \mathbb{1}_2^{\otimes r+2})\mathcal{U}_\mathbf{B}) + H.c.. \tag{A15}$$

Here, $H.c.$ denotes the conjugate transpose of the first term. This operator is not yet unitary because $|0\rangle\langle 0|^{\otimes n}$ is not unitary (it is a projector). However, it is simple to construct a block encoding for a projector as follows. We bring one additional ancilla and implement a conditional unitary operation on the state of the ancilla that is

$$|+\rangle\langle +| \otimes (2\,|0\rangle\langle 0|^{\otimes n} - \mathbb{1}_N) + |-\rangle\langle -| \otimes \mathbb{1}_N . \tag{A16}$$

Because $\langle 0|+\rangle\langle +|\,0\rangle = \langle 0|-\rangle\langle -|\,0\rangle = 1/2$, this gives

$$\langle 0| \left(|+\rangle\langle +| \otimes (2\,|0\rangle\langle 0|^{\otimes n} - \mathbb{1}_N) + |-\rangle\langle -| \otimes \mathbb{1}_N \right) |0\rangle = |0\rangle\langle 0|^{\otimes n} . \tag{A17}$$

Applying $\langle 0| \cdots |0\rangle$ to this operator gives the block encoding

$$\langle 0| \left(|+\rangle\langle +| \otimes (2\,|0\rangle\langle 0|^{\otimes n} - \mathbb{1}_N) + |-\rangle\langle -| \otimes \mathbb{1}_N \right) |0\rangle = |0\rangle\langle 0|^{\otimes n} , \tag{A18}$$

which implements the desired projector. We write $\mathcal{U}_\text{cond}$ for this unitary when acting on the system of $2n+1$ qubits (the space associated with $\mathbf{H}$) plus $r+3$ ancilla qubits. Hence, our block encoding for $\mathbf{H}$ is

$$\frac{1}{\sqrt{2\aleph d}}\mathbf{H} = \langle 0|^{\otimes r+3}\,\mathcal{U}_\mathbf{H}\,|0\rangle^{\otimes r+3} , \tag{A19}$$

where

$$\mathcal{U}_\mathbf{H} := -\mathcal{U}_\text{cond}(|0\rangle\langle 1| \otimes \mathbb{1}_2 \otimes \mathcal{U}_\mathbf{B}) + H.c.. \tag{A20}$$

Here $\mathbb{1}_2$ acts on the extra qubit used for block encoding the projector.

Simulating $\mathcal{U}_\mathbf{H}$ with a quantum circuit requires applying a controlled version of $\mathcal{U}_\mathbf{B}$ and $\mathcal{U}_\mathbf{B}^\dagger$ once, in addition to a simple X gate on the controlled qubit. Simulating $\mathcal{U}_\text{cond}$ can be done with gate complexity $\mathcal{O}(n)$, since it requires applying a conditional phase on $|0\rangle^{\otimes n}$.

$\square$

Appendix B: Oracle lower bound and proof of Lemma 7

Consider the glued-trees oscillator network in Fig. VIA. Let any node represent a unit mass (i.e., $m_j = m_\text{max} = m_\text{min} = 1$ for all $j \in [N]$) and each edge represent a spring of constant 1 (i.e., $\kappa_{jk} = \kappa_\text{max} = 1$ if

(j, k) is an edge and $\kappa_{jk} = 0$ otherwise). The ENTRANCE and EXIT masses are additionally connected to a wall each with a spring of constant also 1. We write $\vec{x}(t) \in \mathbb{R}^N$ and $\dot{\vec{x}}(t) \in \mathbb{R}^N$ for the positions and velocities the masses, where we assume their motion is constrained to one spatial dimension, such as the "horizontal" direction. Although we do not know the actual names of the masses a priori, our labels are such that $j = 1$ refers to the ENTRANCE, $j = 2, 3$ refer to the masses in the second column, and so on, until $j = N = 2^{n+1} - 2$ represents the EXIT mass. The energy of the system is $E = T(t) + U(t)$, where

$$T(t) = \frac{1}{2} \sum_{j=1}^{N} (\dot{x}_j(t))^2 , \tag{B1}$$

$$U(t) = \frac{1}{2} \sum_{j,k>j} \kappa_{jk} (x_j(t) - x_k(t))^2 + \frac{1}{2}\kappa_{11}(x_1(t))^2 + \frac{1}{2}\kappa_{NN}(x_N(t))^2 , \tag{B2}$$

are the kinetic and potential energies, respectively. Newton's equation gives then a set of N coupled second-order differential equations:

$$\ddot{\vec{x}}(t) = -\mathbf{A}\vec{x}(t) , \tag{B3}$$

where $\mathbf{A}$ is the $N \times N$-dimensional symmetric matrix

$$\mathbf{A} = \begin{pmatrix} 3 & -1 & -1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ -1 & 3 & 0 & -1 & \cdots & \cdots & \cdots & \cdots & 0 \\ -1 & 0 & 3 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & -1 & 0 & 3 & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & 3 & 0 & -1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 3 & 0 & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & -1 & 0 & 3 & -1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & -1 & 3 \end{pmatrix} . \tag{B4}$$

Note that $\mathbf{A} = 3I_N - A$, where A is the adjacency matrix of the graph constructed from the two binary trees randomly glued, if we disregard the edges that connect the roots to their respective walls, and I_N is the $N \times N$ identity matrix. Also, $\mathbf{A}$ is positive semi-definite because it is symmetric and diagonally dominant. This property also follows from the potential being $U(t) = \frac{1}{2}\vec{x}(t)^T \mathbf{A}\vec{x}(t) \geq 0$ for all $\vec{x}(t) \in \mathbb{R}^N$, which implies $\mathbf{A} \succeq 0$. In fact, $\mathbf{A}$ is positive definite: The matrix $\mathbf{A}' = \mathbf{A} - |1\rangle\langle 1| - |N\rangle\langle N|$ represents the above network of oscillators where the masses at the roots are not connected to any wall. Hence, $\mathbf{A}'$ contains a non-degenerate eigenvector of eigenvalue zero corresponding to the translations of the system, i.e., the eigenvector $|u_1\rangle := \frac{1}{\sqrt{N}} \sum_j |j\rangle$. The only way for $\mathbf{A}$ to contain an eigenvalue zero is if this eigenvector $|u_1\rangle$ was also an eigenvector of $|1\rangle\langle 1| + |N\rangle\langle N|$, which is not the case. Nevertheless, while $\mathbf{A} \succ 0$, its smallest eigenvalue is exponentially small in n since the expectation $\langle u_1| \mathbf{A} |u_1\rangle = 2/N$ is an upper bound on the smallest eigenvalue.

We would like to show the following property. If $\vec{x}(0) = (0, 0, \ldots, 0)^T$ and $\dot{\vec{x}}(0) = (1, 0, \ldots, 0)^T$, then $\dot{\vec{x}}(t)$ is such that the magnitude of its Nth entry, corresponding to EXIT, is at least polynomially small in n for a time t that is at most polynomial in n (i.e., the kinetic energy of the Nth oscillator is $\Omega(1/\text{poly}(n))$). For these initial conditions, the solution to Eq. (B3) implies

$$\dot{\vec{x}}(t) = \cos\left(t\sqrt{\mathbf{A}}\right) \dot{\vec{x}}(0). \tag{B5}$$

The matrix $\mathbf{A}$, and hence $\sqrt{\mathbf{A}}$, possesses a symmetry that allows one to simulate the dynamics of the N oscillators by considering that of $2n$ oscillators in one spatial dimension, instead. (A similar idea was used in

Ref. [26] to prove that the quantum algorithm solves the problem efficiently.) To show this, we define $2n$ real components:

$$z_l(t) = \frac{1}{\sqrt{N_l}} \sum_{j \in l^{\text{th}} \text{ column}} x_j(t) , \tag{B6}$$

where N_l is the number of masses in the lth column and $l \in [2n]$; that is $N_l = 2^{l-1}$ if $l \le n$ and $N_l = 2^{2n-l}$ if $l \ge n + 1$. Then, for the masses in any column that is not the ENTRANCE ($l = 1$), EXIT ($l = 2n$), or the randomly glued ones where $l = n$ and $l = n + 1$, simple manipulations of Eq. (B3) give

$$\ddot{z}_l(t) = \sqrt{2}z_{l-1}(t) - 3z_l(t) + \sqrt{2}z_{l+1}(t) . \tag{B7}$$

For the masses in the remaining columns, Eq. (B3) gives

$$\ddot{z}_1(t) = -3z_1(t) + \sqrt{2}z_2(t) \tag{B8}$$
$$\ddot{z}_n(t) = \sqrt{2}z_{n-1}(t) - 3z_n(t) + 2z_{n+1}(t) \tag{B9}$$
$$\ddot{z}_{n+1}(t) = 2z_n(t) - 3z_{n+1}(t) + \sqrt{2}z_{n+2}(t) \tag{B10}$$
$$\ddot{z}_{2n}(t) = \sqrt{2}z_{2n-1}(t) - 3z_{2n}(t) . \tag{B11}$$

Then, if $\vec{z}(t) := (z_1(t), \ldots, z_{2n}(t))^T \in \mathbb{R}^{2n}$, Eq. (B3) implies

$$\ddot{\vec{z}}(t) = -\tilde{\mathbf{A}}\vec{z}(t) , \tag{B12}$$

where $\tilde{\mathbf{A}}$ is a $(2n) \times (2n)$ real-symmetric and tridiagonal matrix that, in bra-ket notation, can be written as

$$\tilde{\mathbf{A}} = 3\sum_{l=1}^{2n} |l\rangle\langle l| - \sqrt{2}\left(\sum_{l=1}^{n-1}(|l\rangle\langle l+1| + |l+1\rangle\langle l|) + \sum_{l=n+1}^{2n-1}(|l\rangle\langle l+1| + |l+1\rangle\langle l|)\right)$$
$$- 2(|n\rangle\langle n+1| + |n+1\rangle\langle n|) . \tag{B13}$$

Hence, we have effectively reduced the dimension of the system from N to $2n$ when symmetries are considered. The matrix $\tilde{\mathbf{A}}$ is $3I_{2n} - \tilde{A}$, where $\tilde{A}$ is the $2n \times 2n$ adjacency matrix of the graph constructed from the glued binary trees in the new coordinate system defined above (see Fig. 3). The matrix $\tilde{\mathbf{A}}$ is also positive, with its smallest eigenvalue being exponentially small in n, and any two eigenvalues separated by a gap that is at least $\Delta' = \Omega(1/n^3)$. In particular, if $\vec{z}(0) = (0, 0, \ldots, 0)^T$, the solution to Eq. (B12) implies

$$\dot{\vec{z}}(t) = \cos\left(t\sqrt{\tilde{\mathbf{A}}}\right)\dot{\vec{z}}(0) . \tag{B14}$$

We are interested in the magnitude of $\dot{z}_{2n}(t) \equiv \dot{x}_N(t)$ or, more specifically, the kinetic energy of the Nth oscillator. Similar to Ref. [26], rather than considering a fixed t, we will consider an average over t as follows:

$$P_{\text{EXIT}}(T) := \frac{1}{T}\int_0^T dt |\dot{x}_N(t)|^2 = \frac{1}{T}\int_0^T dt |\dot{z}_{2n}(t)|^2 . \tag{B15}$$

Here, $T > 0$ is set below and $P_{\text{EXIT}}(T)$ is the average kinetic energy of the Nth oscillator from time $t = 0$ to $t = T$ renormalized by the energy $E = 1/2$ for the above initial conditions; for our algorithm, $P_{\text{EXIT}}(T)$

FIG. 3. The $2n \times 2n$ matrix $\tilde{A} = 3I_{2n} - \tilde{\mathbf{A}}$ viewed as the adjacency matrix of a weighted graph.

coincides with the average probability of projecting $|\psi(t)\rangle$ into a basis state that correspond to the EXIT mass. Thus, the goal reduces to showing that there exists $T = \mathcal{O}(\mathrm{poly}(n))$ such that $P_{\mathrm{EXIT}}(T) = \Omega(1/\mathrm{poly}(n))$. If we prove this, then Eq. (B15) automatically implies the existence of $t = \mathcal{O}(\mathrm{poly}(n))$ such that $K_N(t) = \Omega(1/\mathrm{poly}(n))$, which is our main goal. Finding such t can be done classically by simulating Eq. (B12) in time polynomial in n.

We can write

$$\dot{z}_{2n}(t) = \langle 2n| \frac{1}{2} \left(e^{it\sqrt{\tilde{\mathbf{A}}}} + e^{-it\sqrt{\tilde{\mathbf{A}}}} \right) |1\rangle \ . \tag{B16}$$

Let $\{|\lambda_1\rangle, \ldots, |\lambda_{2n}\rangle\}$ be the $2n$ eigenvectors of the adjacency matrix $\tilde{A}$ of eigenvalues $\lambda_1 < \lambda_2 < \ldots < \lambda_{2n}$. These are also eigenvectors of $\sqrt{\tilde{\mathbf{A}}}$ of eigenvalues $\gamma_l := \sqrt{3 - \lambda_l} > 0$. This implies

$$
\begin{aligned}
P_{\mathrm{EXIT}}(\infty) &:= \lim_{T \to \infty} P_{\mathrm{EXIT}}(T) \\
&= \lim_{T \to \infty} \frac{1}{T} \left(\int_0^T dt \, \langle 2n| \frac{1}{2} \left(e^{it\sqrt{\tilde{\mathbf{A}}}} + e^{-it\sqrt{\tilde{\mathbf{A}}}} \right) |1\rangle\langle 1| \frac{1}{2} \left(e^{it\sqrt{\tilde{\mathbf{A}}}} + e^{-it\sqrt{\tilde{\mathbf{A}}}} \right) |2n\rangle \right) \\
&= \sum_{l,l'=1}^{2n} \langle 2n|\lambda_l\rangle\langle\lambda_l|1\rangle\langle 1|\lambda_{l'}\rangle\langle\lambda_{l'}|2n\rangle \lim_{T \to \infty} \frac{1}{T} \int_0^T dt \, \cos(t\gamma_l)\cos(t\gamma_{l'}) \\
&= \frac{1}{2} \sum_{l=1}^{2n} \langle 2n|\lambda_l\rangle\langle\lambda_l|1\rangle\langle 1|\lambda_l\rangle\langle\lambda_l|2n\rangle \\
&= \frac{1}{2} \sum_{l=1}^{2n} |\langle\lambda_l|1\rangle|^2 \, |\langle\lambda_l|2n\rangle|^2 \\
&= \frac{1}{2} \sum_{l=1}^{2n} |\langle\lambda_l|1\rangle|^4 \\
&\geq \frac{1}{2} \frac{1}{2n} \sum_{l=1}^{2n} |\langle\lambda_l|1\rangle|^2 \\
&= \frac{1}{4n} \ . \tag{B17}
\end{aligned}
$$

To obtain this we used $\lim_{T \to \infty} \frac{1}{T} \int_0^T dt \, e^{it(\gamma_l - \gamma_{l'})} = \delta_{l,l'}$, $\lim_{T \to \infty} \frac{1}{T} \int_0^T dt \, e^{it(\gamma_l + \gamma_{l'})} = 0$ since the eigenvalues are positive, $|\langle\lambda_l|1\rangle| = |\langle\lambda_l|2n\rangle|$ for all $l \in [2n]$ due to a reflection symmetry of the network (i.e., the adjacency matrix of the graph in the line is invariant under the transformation $l \leftrightarrow 2n - l + 1$), and also the Cauchy Schwarz inequality for the last line as there are $2n$ different eigenvectors. This would already prove the desired result, but we are interested in finite times $T < \infty$ and, more precisely, showing a similar bound for $T = \mathcal{O}(\mathrm{poly}(n))$.

Using again the reflection symmetry of the network, for $T < \infty$ the average success probability can be

written as

$$P_{\text{EXIT}}(T) =$$

$$= \sum_{l,l'} \langle 2n | \lambda_l \rangle \langle \lambda_l | 1 \rangle \langle 1 | \lambda_{l'} \rangle \langle \lambda_{l'} | 2n \rangle \frac{1}{T} \int_0^T dt \cos(t\gamma_l) \cos(t\gamma_{l'})$$

$$= \sum_{l,l'} |\langle 1 | \lambda_l \rangle|^2 |\langle 1 | \lambda_{l'} \rangle|^2 \frac{1}{T} \int_0^T dt \cos(t\gamma_l) \cos(t\gamma_{l'})$$

$$= \frac{1}{2} \sum_l |\langle 1 | \lambda_l \rangle|^4 \frac{1}{T} \int_0^T dt (1 + \cos(2t\gamma_l)) + \sum_{l \neq l'} |\langle 1 | \lambda_l \rangle|^2 |\langle 1 | \lambda_{l'} \rangle|^2 \frac{1}{T} \int_0^T dt \cos(t\gamma_l) \cos(t\gamma_{l'})$$

$$= \frac{1}{2} \sum_l |\langle 1 | \lambda_l \rangle|^4 + \frac{1}{2} \sum_l |\langle 1 | \lambda_l \rangle|^4 \frac{1}{T} \int_0^T dt \cos(2t\gamma_l) + \sum_{l \neq l'} |\langle 1 | \lambda_l \rangle|^2 |\langle 1 | \lambda_{l'} \rangle|^2 \frac{1}{T} \int_0^T dt \cos(t\gamma_l) \cos(t\gamma_{l'})$$

$$= P_{\text{EXIT}}(\infty) + \frac{1}{2} \sum_l |\langle 1 | \lambda_l \rangle|^4 \frac{1}{T} \int_0^T dt \cos(2t\gamma_l) + \sum_{l \neq l'} |\langle 1 | \lambda_l \rangle|^2 |\langle 1 | \lambda_{l'} \rangle|^2 \frac{1}{T} \int_0^T dt \cos(t\gamma_l) \cos(t\gamma_{l'}) , \quad \text{(B18)}$$

where we used the identity $\cos^2(\alpha) = \frac{1}{2}(1 + \cos(2\alpha))$. Then, the correction to $P_{\text{EXIT}}(\infty)$ for $T < \infty$ is such that

$$|P_{\text{EXIT}}(T) - P_{\text{EXIT}}(\infty)| = \left| \sum_{l \neq l'} |\langle 1 | \lambda_l \rangle|^2 |\langle 1 | \lambda_{l'} \rangle|^2 \frac{1}{T} \int_0^T dt \cos(t\gamma_l) \cos(t\gamma_{l'}) + \right.$$

$$\left. \frac{1}{2} \sum_l |\langle 1 | \lambda_l \rangle|^4 \frac{1}{T} \int_0^T dt \cos(2t\gamma_l) \right| . \quad \text{(B19)}$$

Note that, for $l \neq l'$,

$$\left| \frac{1}{T} \int_0^T dt \cos(t\gamma_l) \cos(t\gamma_{l'}) \right| = \frac{1}{T} \left| \frac{1}{2} \left(\frac{\sin(T(\gamma_l + \gamma_{l'}))}{\gamma_l + \gamma_{l'}} + \frac{\sin(T(\gamma_l - \gamma_{l'}))}{\gamma_l - \gamma_{l'}} \right) \right| \quad \text{(B20)}$$

$$\leq \frac{1}{T\Delta} , \quad \text{(B21)}$$

where Δ is the minimum spectral gap between any pair of eigenvalues γ_l, that is, $\Delta := \min_l |\gamma_{l+1} - \gamma_l|$. (Also $|\gamma_l + \gamma_{l'}| > \Delta$ if $l \neq l'$.) Also, if we order the eigenvalues so that $\gamma_{l+1} > \gamma_l$ for all $l = [2n-1]$, for $l \geq 2$ we have $\gamma_l \geq \Delta$ and

$$\left| \frac{1}{T} \int_0^T dt \cos(2t\gamma_l) \right| = \frac{1}{T} \left| \frac{\sin(2t\gamma_l)}{2\gamma_l} \right| \leq \frac{1}{2T\Delta}. \quad \text{(B22)}$$

For $l = 1$, the eigenvalue γ_l is exponentially small in n and the corresponding average can be $\mathcal{O}(1)$. Combining these equations and using the reflection symmetry of the network we obtain

$$|P_{\text{EXIT}}(T) - P_{\text{EXIT}}(\infty)| \leq \frac{1}{T\Delta} \sum_{l \neq l'} |\langle \lambda_l | 1 \rangle|^2 |\langle \lambda_{l'} | 1 \rangle|^2 + \frac{1}{4T\Delta} \sum_{l \geq 2} |\langle \lambda_l | 1 \rangle|^4 + \frac{1}{2} |\langle \lambda_1 | 1 \rangle|^4$$

$$\leq \frac{1}{T\Delta} \sum_{l,l'} |\langle \lambda_l | 1 \rangle|^2 |\langle \lambda_{l'} | 1 \rangle|^2 + \frac{1}{2} |\langle 1 | \lambda_1 \rangle|^4$$

$$\leq \frac{1}{T\Delta} + \frac{1}{2} |\langle 1 | \lambda_1 \rangle|^4 , \quad \text{(B23)}$$

where we also used $\sum_l |\langle 1 | \lambda_l \rangle|^2 = 1$. We need to show that both these terms are small.

In Ref. [26] it has been shown that the eigenvalues λ_l's are separated by spectral gaps bounded by $\Delta' = \Omega(1/n^3)$. The same follows for the γ_l's. More precisely, suppose that the two closest eigenvalues of $\tilde{\mathbf{A}}$ are $3 - \lambda \leq 6$ and $3 - (\lambda + \Delta')$. Then, the two closest eigenvalues of $\sqrt{\tilde{\mathbf{A}}}$ are separated as

$$
\begin{aligned}
\Delta &= \sqrt{3 - \lambda} - \sqrt{3 - (\lambda + \Delta')} \\
&= \sqrt{3 - \lambda}\left(1 - \sqrt{1 - \frac{\Delta'}{3 - \lambda}}\right) \\
&\geq \sqrt{3 - \lambda}\left(1 - \left(1 - \frac{\Delta'}{3 - \lambda}\right)\right) \\
&\geq \frac{\Delta'}{\sqrt{3 - \lambda}} \\
&\geq \frac{\Delta'}{\sqrt{6}} \; .
\end{aligned}
\tag{B24}
$$

Then, since $\Delta' = \Omega(1/n^3)$, we have $\Delta = \Omega(1/n^3)$. This implies that we can choose $T = \mathcal{O}(n^4)$ so that, for example,

$$
\frac{1}{T\Delta} \leq \frac{1}{8n} \; .
\tag{B25}
$$

It is also possible to show that $|\langle 1 | \lambda_1 \rangle|^4$ is exponentially small in n. Consider the vector

$$
|v_1\rangle := \sum_{l=1}^{n} \frac{1}{\sqrt{2^{2+n-l}}} |l\rangle + \sum_{l=n+1}^{2n} \frac{1}{\sqrt{2^{1+l-n}}} |l\rangle \; .
\tag{B26}
$$

The length of this vector is $\sqrt{\sum_{l=1}^{n}(1/2^{1+n-l})} = \sqrt{1 - 1/2^n}$ being exponentially close to 1. In addition,

$$
\begin{aligned}
\tilde{\mathbf{A}} |v_1\rangle &= \\
3|v_1\rangle &- \sqrt{2}\left(\sum_{l=1}^{n-1} \frac{1}{\sqrt{2^{2+n-l}}} |l+1\rangle + \sum_{l=2}^{n} \frac{1}{\sqrt{2^{2+n-l}}} |l-1\rangle + \sum_{l=n+1}^{2n-1} \frac{1}{\sqrt{2^{1+l-n}}} |l+1\rangle + \sum_{l=n+2}^{2n} \frac{1}{\sqrt{2^{1+l-n}}} |l-1\rangle\right) \\
&- 2\frac{1}{2}(|n+1\rangle + |n\rangle) \\
&= 3|v_1\rangle - \sqrt{2}\left(\sum_{l=2}^{n} \frac{1}{\sqrt{2^{3+n-l}}} |l\rangle + \sum_{l=1}^{n-1} \frac{1}{\sqrt{2^{1+n-l}}} |l\rangle + \sum_{l=n+2}^{2n} \frac{1}{\sqrt{2^{l-n}}} |l\rangle + \sum_{l=n+1}^{2n-1} \frac{1}{\sqrt{2^{2+l-n}}} |l\rangle\right) \\
&- 2\frac{1}{2}(|n+1\rangle + |n\rangle) \\
&= 3|v_1\rangle - 3\sum_{l=2}^{n-1} \frac{1}{\sqrt{2^{2+n-l}}} |l\rangle - \frac{1}{2}|n\rangle - \frac{1}{\sqrt{2^{n-1}}} |1\rangle - 3\sum_{l=n+2}^{2n-1} \frac{1}{\sqrt{2^{1+l-n}}} |l\rangle - \frac{1}{\sqrt{2^{n-1}}} |2n\rangle - \frac{1}{2}|n+1\rangle \\
&- 2\frac{1}{2}(|n+1\rangle + |n\rangle) \\
&= 3|v_1\rangle - 3|v_1\rangle + \frac{1}{2\sqrt{2^{n-1}}} |1\rangle + \frac{1}{2\sqrt{2^{n-1}}} |2n\rangle \\
&= \frac{1}{2\sqrt{2^{n-1}}} |1\rangle + \frac{1}{2\sqrt{2^{n-1}}} |2n\rangle \; .
\end{aligned}
\tag{B27}
$$

This implies, $\|\tilde{\mathbf{A}}(|v_1\rangle / \||v_1\rangle\|)\| = \mathcal{O}(1/\sqrt{2^n})$ and that $|v_1\rangle / \||v_1\rangle\|$ is exponentially close to the normalized eigenvector of lowest eigenvalue as the spectral gaps are at least $\Delta = \Omega(1/n^3)$; that is $\|(|v_1\rangle / \||v_1\rangle\|) - |\lambda_1\rangle\| =$
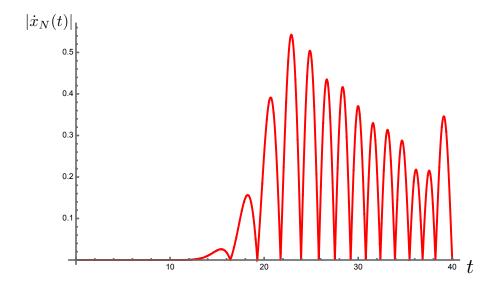
FIG. 4. Numerical simulation of the network in Fig. VI A for $n = 20$ ($N = 2^{21} - 2$). Initially $\vec{x}(0) = (0, 0, \ldots, 0)^T$ and $\dot{\vec{x}}(0) = (1, 0, \ldots, 0)^T$. At time $t \approx 2n$, $|\dot{x}_N(t)|$ becomes significant. Similar behavior is observed for larger values of n.

$\mathcal{O}(1/\exp(n))$. Since the first entry of $|v_1\rangle / \| |v_1\rangle \|$ is $\mathcal{O}(1/\exp(n))$, these results imply $|\langle 1|\lambda_1\rangle| = \mathcal{O}(1/\exp(n))$. Combining these bounds, and for the choice of T above, we obtain

$$|P_{\text{EXIT}}(T) - P_{\text{EXIT}}(\infty)| \leq \frac{1}{8n} + \mathcal{O}(1/\exp(n)) . \tag{B28}$$

Together with Eq. (B17), this gives

$$P_{\text{EXIT}}(T) \geq \frac{1}{8n} - \mathcal{O}(1/\exp(n)) , \tag{B29}$$

which is the desired result: we can choose $T = \mathcal{O}(n^4)$ so that the average probability $P_{\text{EXIT}}(T)$ is $\Omega(1/n)$, implying the existence of $t \in [0, T]$ with the desired property $|\dot{x}_N(t)| = \Omega(1/\text{poly}(n))$. $\qquad\square$

We note that numerical solutions show a stronger result, where $t \propto n$ suffices. See Fig. 4 for an example.

Appendix C: BQP-completeness and proof of Thm. 3

In this section our goal is to show Thm. 3, which establishes that Problem 3 is BQP-complete. A problem is BQP-complete if it is in BQP and if it is BQP-hard, which means that every problem in BQP can be reduced to it by classical polynomial-time reductions. Since we have already established that Problem 3 is in BQP (a consequence of Thm. 1), we only need to show it is BQP-hard.

To show our problem is BQP-hard, we start the reduction from the following BQP-complete problem:

Problem 7. *Given a quantum circuit on q qubits with $L = \text{poly}(q)$ gates over the gate set of Hadamard, Pauli X, and Toffoli acting on the initial state $|0\rangle^{\otimes q}$, decide if the output state has overlap at least $1 - 1/\exp(\sqrt{q})$ with $|0\rangle^{\otimes q}$ or has overlap at most $1/\exp(\sqrt{q})$ with $|0\rangle^{\otimes q}$, promised that one of these is the case.*

This problem is easily seen to be in BQP. It is BQP-hard by reduction from the standard BQP-complete problem, which has a similar input, but the goal is to decide a different property of the output state: Upon

measuring the first qubit of the output state, we have to decide if it is 1 with probability at least $2/3$ or 1 with probability at most $1/3$, promised that one of these is the case. By known results, we can assume our gate set is real and contains only Hadamard and Toffoli, since this is a universal gate set for real quantum computation, which is computationally as powerful as complex quantum computation [39, 40]. We also add the single-qubit Pauli X gate so that we can create the state $|1\rangle$ from $|0\rangle$, as we want the computation to start with all qubits in state $|0\rangle$. Also, the X and Toffoli gates generate swap gates, and we can then assume that all Hadamards act on the same qubit. This property will be useful to simplify the presentation of the proof, but it is not necessary, and one could in principle allow Hadamards to act on any qubit.

Given such a circuit, we can amplify the constants $2/3$ and $1/3$ to $1 - 1/\exp(q)$ and $1/\exp(q)$ by running $\mathcal{O}(q)$ copies of the circuit in parallel and taking the majority vote of the first output qubits of all circuits. Now we have a new circuit on $\mathcal{O}(q^2)$ qubits whose first qubit is almost certainly (with probability $1 - 1/\exp(q)$) $|1\rangle$ when the answer is yes and almost certainly $|0\rangle$ when the answer is no. We then use an additional qubit and copy this answer to that qubit. Since the qubit we are copying is exponentially close to being $|0\rangle$ or $|1\rangle$, let us assume it is one of these, and this will only introduce an error of $1/\exp(q)$. We then run the circuit's inverse on the remaining qubits to restore them to the all zero state. The resulting output state is now exponentially close to all zeros if the additional qubit was also zero, which happens when the input was a no instance. If it was a yes instance, the additional qubit we added will be exponentially close to $|1\rangle$, and hence the overall state will have almost no amplitude on the all zeros state.

Thus deciding if the output state of a $\mathcal{O}(q^2)$ qubit circuit of size $\text{poly}(q)$ is $1 - 1/\exp(q)$ close to the all zeros state or has at most $1/\exp(q)$ overlap on the all zeros state is BQP-complete. This yields the stated result by renaming q to q^2. (Our proof below does not require the closeness to be exponentially small, and it would suffice to have inverse polynomial closeness, as long as this polynomial was smaller than all the other polynomials appearing in the problem.)

1. From a circuit to a network of oscillators

We now show that our quantum algorithm for simulating coupled classical oscillators allows us to solve the BQP-complete problem above. We begin with the given quantum circuit on q qubits with $L = \text{poly}(q)$ gates. If the gates are U_1 to U_L, the output of this circuit is $U_L \ldots U_1 |0\rangle^{\otimes q}$. Our goal is to decide if this state is essentially $|0\rangle^{\otimes q}$ or has essentially no overlap with $|0\rangle^{\otimes q}$. The unitaries U_1 to U_L are either the single-qubit gates Hadamard H or Pauli X (tensored with identity on all other qubits) or the three-qubit Toffoli gate Toff (tensored with identity on all other qubits). These gates are

$$H = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}, \qquad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \text{and} \qquad \text{Toff} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \qquad \text{(C1)}$$

in the corresponding one-qubit and three-qubit subspaces. It will be very useful in our context that these are real as the resulting Hamiltonian will have real entries, which we require because the κ_{jk} are real.

From this circuit we will create a network of N oscillators, where $N = (L+1)2^{q+1}$. It is convenient to label each oscillator $j \in [N]$ using two indexes $j \to (l, r)$, where $l \in [L+1]$, $r \in [2^{q+1}]$, and $j = (l-1)2^{q+1} + r$. We will call the oscillator with $l = L+1$ and $r = 1$ the output oscillator. In our construction, the $N \times N$ matrix of spring constants $\mathbf{K}$ that describes the couplings between oscillators, which depends on the U_l's, is 4-sparse and each entry is bounded as $\kappa_{jk} \leq 4$. We will have all masses be $m_j = 1$. Note that $\mathbf{A} = \sqrt{\mathbf{M}}^{-1} \mathbf{F} \sqrt{\mathbf{M}}^{-1} = \mathbf{F}$ and $\vec{y}(t) = \sqrt{\mathbf{M}} \vec{x}(t) = \vec{x}(t)$ in this case. This simplifies our analysis.

In the following, we show that for our oscillator network, there exists a time $t = \text{polylog}(N)$ such that, for the initial conditions where $\vec{x}(0) = (0, \ldots, 0)^T$, $\dot{x}_1(0) = 1$, $\dot{x}_2(0) = -1$, and $\dot{x}_j(0) = 0$ for all $j > 2$, the kinetic energy of the output oscillator is either exponentially close to 0 (when the original circuit had almost no

overlap with $|0\rangle^{\otimes q}$), or $1/\mathrm{polylog}(N)$ (when the original circuit's output is very close to $|0\rangle^{\otimes q}$). Since our algorithm allows us to estimate the kinetic energy of a given oscillator to additive $1/\mathrm{polylog}(N)$ precision, we can distinguish these two cases with complexity $\mathcal{O}(\mathrm{polylog}(N))$. The initial quantum state $|\psi(0)\rangle$ is simply $|-\rangle$ tensored with the all zeros state, and hence easy to prepare. This proves that Problem 2 is BQP-complete even with these constraints on the initial conditions, t, ϵ, $\mathbf{M}$, and $\mathbf{K}$, which is the desired result.

We now provide the details of this construction. We use the bra-ket notation for specifying vectors and matrices, which is convenient for relating properties of the classical system with properties of quantum states. We will specify our oscillator network using the $\mathbf{A}$ matrix, instead of the $\mathbf{K}$ matrix. This real matrix is PSD and has non-negative entries on the diagonal and non-positive entries on the off-diagonal. The standard Feynman-Kitaev [28, 29] circuit-to-Hamiltonian construction gives us a Hamiltonian of the form

$$\sum_{l=1}^{L}(|l\rangle\langle l+1| + |l+1\rangle\langle l|) \otimes W_l\,, \tag{C2}$$

where W_l is the lth gate in the circuit, i.e., U_l. We cannot let $\mathbf{A}$ equal this Hamiltonian, since the matrix is not PSD and there are off-diagonal entries of both signs. The first issue is easily fixed by adding a multiple of the identity, but it will require some work to fix the second issue. We will adjoin an additional qubit to work in a slightly larger space. Formally, we consider the following Hilbert space:

$$\mathcal{H} = \mathcal{H}_{\mathrm{clock}} \otimes \mathcal{H}_{\mathrm{comp}} \otimes \mathcal{H}_2\,. \tag{C3}$$

Here, $\mathcal{H}_{\mathrm{clock}}$ is the $(L+1)$-dimensional Hilbert space that is used to describe the state of the clock (i.e., corresponding to the first register in Eq. (C2)), which is used to track the progress of a simulated quantum computation on a state in $\mathcal{H}_{\mathrm{comp}} \otimes \mathcal{H}_2$ where $\mathcal{H}_{\mathrm{comp}}$ is 2^q-dimensional and $\mathcal{H}_2$ is two-dimensional. The purpose of $\mathcal{H}_{\mathrm{comp}}$ is to store the state of the given quantum circuit at a time given by the clock. The purpose of $\mathcal{H}_2$ is to address the issue on the off-diagonal entries raised above. The off-diagonal entries of $\mathbf{A}$ need to have the same sign, but our gate set includes a gate, H, with entries of both signs. We address this by providing a resource state $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ in the last register which we use to effectively create negative signs in a subspace although the operators will only have positive entries. Similar constructions that create Hamiltonians with non-negative entries have been used within the context of Hamiltonian complexity; see Refs. [30, 31] for example.

We will map our circuit to a system of oscillators that satisfy the assumptions made in Problem 1 by choosing the couplings to implement an "encoded" sequence of gates $\{W_l\}_l$ as

$$\mathbf{A} = 4\mathbb{1}_N - \sum_{l=1}^{L}(|l\rangle\langle l+1| + |l+1\rangle\langle l|) \otimes W_l, \tag{C4}$$

where each W_l corresponds to an encoded version of the lth gate in the original circuit, which is either a Hadamard, X, or a Toffoli, and $\mathbb{1}_N$ is the identity matrix on the system of dimension N. Notice that W_l lives in $\mathcal{H}_{\mathrm{comp}} \otimes \mathcal{H}_2$, whereas the original unitaries U_l lived in $\mathcal{H}_{\mathrm{comp}}$. So we encode each gate into a larger space with 1 additional qubit.

We want all the entries in the encoded versions of Hadamard, X, and Toffoli to be non-negative, which will make the off-diagonal entries non-positive in $\mathbf{A}$ due to Eq. (C4). Let's start with the Hadamard matrix. Without loss of generality, because we can swap qubits, we assume the Hadamard always acts on qubit q, the last qubit of $\mathcal{H}_{\mathrm{comp}}$. We describe our encoded Hadamard gate via the following positive-valued real-symmetric block matrix acting on the last qubit of $\mathcal{H}_{\mathrm{comp}}$ and $\mathcal{H}_2$:

$$H_{\mathrm{enc}} = \frac{1}{\sqrt{2}} \begin{pmatrix} \mathbb{1}_2 & \mathbb{1}_2 \\ \mathbb{1}_2 & X \end{pmatrix}. \tag{C5}$$

This can be seen to act as a logical Hadamard when acting on a state of the form $|\psi\rangle|-\rangle$ because of the

following block-encoding result:

$$
\begin{aligned}
(\mathbb{1}_2 \otimes \langle -|) H_{\mathrm{enc}} (\mathbb{1}_2 \otimes |-\rangle) &= \frac{1}{\sqrt{2}} \left((\mathbb{1}_2 \otimes \langle -|) \begin{pmatrix} \mathbb{1}_2 & \mathbb{1}_2 \\ \mathbb{1}_2 & X \end{pmatrix} (\mathbb{1}_2 \otimes |-\rangle) \right) \\
&= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\
&= H.
\end{aligned}
\tag{C6}
$$

Note here that unlike a conventional block encoding, the encoded Hadamard gate H_{enc} is not a unitary operation. However it block encodes a unitary operation within the logical subspace where the Hilbert space $\mathcal{H}_2$ contains the state $|-\rangle$, which is sufficient for our purposes as it represents the Hadamard gate as a real-symmetric matrix with positive entries.

The Toffoli and X gates have non-negative entries and so we define the encoded Toffoli and X gates on $\mathcal{H}_{\mathrm{comp}} \otimes \mathcal{H}_2$ as simply $U_l \otimes \mathbb{1}_2$. The action of this matrix on the first register ($\mathcal{H}_{\mathrm{comp}}$) is a Toffoli gate or X gate, independent of the second register ($\mathcal{H}_2$). Finally, as in the standard circuit-to-Hamiltonian construction, the initial state that we use is $|l = 1\rangle \otimes |0\rangle^{\otimes q}$ on the first two spaces, and $|-\rangle$ on the last space, as discussed. As mentioned above, the W_l's, which are not necessarily unitary on the entire Hilbert space (but act as unitaries on the subspace where the last register is $|-\rangle$), have non-negative entries $\in \{0, 1/\sqrt{2}, 1\}$ and are of sparsity at most 2.

The matrix $\mathbf{F} = \mathbf{A}$ contains all the desired properties to describe N coupled oscillators. Its off-diagonal entries are $a_{jk} \in \{0, -1, -1/\sqrt{2}\}$, for $j, k \in [N]$, $j \neq k$. This agrees with the assumptions that the off-diagonal elements in $\mathbf{F}$ are negative given in Sec. III. All diagonal entries are $a_{jj} = 4$, which similarly agrees with the requirement in Sec. III that the diagonal elements of $\mathbf{F}$ are positive. The spring constants of the system are such that $\kappa_{jk} = -a_{jk}$ for $j \neq k$ and then $\kappa_{jk} \in \{0, 1/\sqrt{2}, 1\}$. Also, $\kappa_{jj} = a_{jj} - \sum_{k \neq j} \kappa_{jk} = 4 - \sum_{k \neq j} a_{jk}$. The matrix $\mathbf{K}$ has a non-zero diagonal entry and it can have at most 3 nonzero off-diagonal entries. For example, if $W_l = H_{\mathrm{enc}}$ and $W_{l+1} = \mathrm{Toff}$ or $W_{l+1} = X$, then the corresponding row will have at most 3 non-zero matrix elements off the diagonal with coefficients $1/\sqrt{2}$ for two and 1 for the remainder. Maximizing over all such possibilities $\sum_{k \neq j} a_{jk} \leq 1 + 2/\sqrt{2}$ in our construction, we obtain $\kappa_{jj} > 0$. (Since the Hadamard gates act on the same qubit, we do not have two consecutive Hadamards in our circuit.) The matrix of spring constants $\mathbf{K}$ is then 4-sparse and each entry satisfies $\kappa_{jk} \leq 4$. Given $j \in [N]$, it is possible to compute all neighbors $a(j, \ell)$ of j, where $\ell \in [4]$, and the corresponding non-zero spring constants, in polynomial time from the polynomial-sized representation in Eq. (C4). This gives an efficient circuit to query the matrix $\mathbf{K}$ as in Sec. IV. Now that we have established that a query to the elements of the oscillator can be efficiently simulated we will now turn to showing that the classical dynamics under an exponentially large $\mathbf{A}$ can implement an arbitrary quantum computation.

2. Solving the original problem using the dynamics of oscillators

As discussed above, the initial state we want to use is $|l = 1\rangle \otimes |0\rangle^{\otimes q} \otimes |-\rangle$. As a vector, this is a vector with the first entry equal to $+1/\sqrt{2}$, the second entry equal to $-1/\sqrt{2}$, and the remaining entries zero. Using our problem's encoding in (2), this corresponds to the initial state where $\dot{y}_1(0) = 1$, $\dot{y}_2(0) = -1$, $\dot{y}_j(0) = 0$ for all $j > 2$, $\vec{x}(0) = 0$, and $E = 1$. This corresponds to all the oscillators starting at their rest positions, with the first oscillator having velocity $+1$, the second one having velocity -1, and the remaining oscillators being stationary.

When the initial conditions are such that $\vec{x}(0) = (0, \ldots, 0)^T$, the solution to Newton's equations is even easier to describe, and Eq. (11) implies

$$
\dot{\vec{x}}(t) = \dot{\vec{y}}(t) = \mathrm{Re}\left\{ e^{i\sqrt{\mathbf{A}}t} \dot{\vec{y}}(0) \right\} = \cos\left(\sqrt{\mathbf{A}} t\right) \dot{\vec{y}}(0).
\tag{C7}
$$

In the rest of this section we want to understand the behavior of $\dot{\vec{y}}(t)$ for our initial conditions.

We start by defining an $N \times N$ matrix $\mathbf{A}'$, which is obtained from $\mathbf{A}$ in Eq. (C4) by replacing $W_l \to U_l \otimes \mathbb{1}_2$ when U_l is a Hadamard gate (i.e., $\mathbf{A}'$ is allowed to have off-diagonal entries of both signs). We do this for

simplicity to remove the dependence on the $|-\rangle$ state in the ancilla. Specifically let $U_l \in \{H, X, \text{Toff}\}$ and

$$\mathbf{A}' = 4\mathbb{1}_N - \sum_{l=1}^{L} (|l\rangle \langle l+1| + |l+1\rangle \langle l|) \otimes U_l \otimes \mathbb{1}_2 . \tag{C8}$$

The block-encoding property of Eq. (C6) implies

$$\begin{aligned}
\dot{\vec{y}}(t) &= \cos\left(\sqrt{\mathbf{A}}t\right)\dot{\vec{y}}(0) \\
&= \cos\left(\sqrt{\mathbf{A}}t\right)|l=1\rangle \otimes |0\rangle^{\otimes q} \otimes |-\rangle \\
&= \cos\left(\sqrt{\mathbf{A}'}t\right)|l=1\rangle \otimes |0\rangle^{\otimes q} \otimes |-\rangle .
\end{aligned} \tag{C9}$$

That is, W_l and $U_l \otimes \mathbb{1}_2$ act identically on the subspace specified by $|-\rangle$. As a corollary, the velocities of the oscillators are such that the last register remains in $|-\rangle$. The matrix $\mathbf{A}'$ is also 4-sparse and satisfies $\mathbf{A}' \geq 0$, implying that $\sqrt{\mathbf{A}'}$ is a well defined $N \times N$ Hermitian matrix that is row-computable.

We define the $N \times N$ select unitary ($U_0 := \mathbb{1}_{2^q}$)

$$S := \sum_{l=0}^{L} |l+1\rangle\langle l+1| \otimes U_l \dots U_0 \otimes \mathbb{1}_2 , \tag{C10}$$

which implements the unitary in the circuit up to U_l when the first register is $|l+1\rangle$, i.e., a vector in $\mathbb{R}^{L+1}$ with entry 1 in position $l+1$ and zeroes elsewhere. Note that S and $S^\dagger$ leave any vector represented as $|l=1\rangle \otimes |\psi\rangle$, $|\psi\rangle \in \mathbb{C}^{2^{q+1}}$, invariant. We also define the $N \times N$ Hermitian matrix

$$\mathbf{X} := 4\mathbb{1}_N - \sum_{l=1}^{L} (|l\rangle\langle l+1| + |l+1\rangle\langle l|) \otimes \mathbb{1}_{2^{q+1}} . \tag{C11}$$

Then, by inspection of Eq. (C8), we obtain from the fact that each U_l in our gate set is Hermitian and unitary

$$\mathbf{A}' = S\mathbf{X}S^\dagger . \tag{C12}$$

Since $U_0 = \mathbb{1}_{2^q}$, for our initial conditions $\dot{\vec{y}}(0) = |l=1\rangle \otimes |0\rangle^{\otimes q} \otimes |-\rangle$,

$$\begin{aligned}
\dot{\vec{y}}(t) &= \cos\left(\sqrt{\mathbf{A}'}t\right)|l=1\rangle \otimes |0\rangle^{\otimes q} \otimes |-\rangle \\
&= S\cos\left(\sqrt{\mathbf{X}}t\right)S^\dagger |l=1\rangle \otimes |0\rangle^{\otimes q} \otimes |-\rangle \\
&= S\cos\left(\sqrt{\mathbf{X}}t\right)|l=1\rangle \otimes |0\rangle^{\otimes q} \otimes |-\rangle .
\end{aligned} \tag{C13}$$

Since $\mathbf{X}$ acts trivially on $\mathcal{H}_{\text{comp}} \otimes \mathcal{H}_2$, it is useful to define

$$\mathbf{X}' := 4\mathbb{1}_{L+1} - \sum_{l=1}^{L} (|l\rangle\langle l+1| + |l+1\rangle\langle l|) , \tag{C14}$$

implying

$$\begin{aligned}
\dot{\vec{y}}(t) &= S\cos\left(\sqrt{\mathbf{X}}t\right)|l=1\rangle \otimes |0\rangle^{\otimes q} \otimes |-\rangle \\
&= S\left(\cos\left(\sqrt{\mathbf{X}'}t\right)|l=1\rangle\right) \otimes |0\rangle^{\otimes q} \otimes |-\rangle .
\end{aligned} \tag{C15}$$

This compact expression for the vector of velocities is useful to show the desired result.

In general, we can write

$$\cos\left(\sqrt{\mathbf{X}'}t\right)|l=1\rangle = \sum_{l=1}^{L+1} \alpha_l(t)\,|l\rangle\ , \tag{C16}$$

with $\alpha_l(t) \in \mathbb{R}$, because the Taylor expansion of the cosine function has only even powers of $\sqrt{\mathbf{X}'}$. Then

$$\dot{\vec{y}}(t) = S\left(\cos\left(\sqrt{\mathbf{X}'}t\right)|l=1\rangle\right) \otimes |0\rangle^{\otimes q} \otimes |-\rangle$$

$$= \sum_{l=1}^{L+1} \alpha_l(t)\,|l\rangle \otimes (U_{l-1}\ldots U_0\,|0\rangle^{\otimes q}) \otimes |-\rangle\ . \tag{C17}$$

Hence, when the first register in the tensor product is $l = L + 1$, the second register is $U_L\ldots U_1\,|0\rangle^{\otimes q}$, and recall that our goal was to decide if this state was close to or far from $|0\rangle^{\otimes q}$.

So if we measure the output state, in the case where the circuit's output was close to $|0\rangle^{\otimes q}$, we will see $|L+1\rangle\,|0\rangle^{\otimes q}$ with probability $(1 - \frac{1}{\exp(\sqrt{q})})|\alpha_{L+1}(t)|^2 \geq \frac{1}{2}|\alpha_{L+1}(t)|^2$, and if the circuit's output had $1/\exp(\sqrt{q})$ overlap with $|0\rangle^{\otimes q}$, then we'll see $|L+1\rangle\,|0\rangle^{\otimes q}$ with at most $1/\exp(\sqrt{q})$ probability.

So all we have to show is that there is exists a $t = \mathrm{polylog}(N)$ such that $|\alpha_{L+1}(t)|^2 = \Omega(1/\mathrm{polylog}(N)) = \Omega(1/\mathrm{poly}(q))$, which will allow us to distinguish the two cases by using our algorithm $\mathcal{O}(\mathrm{polylog}(N)) = \mathrm{poly}(q)$ times.

3. Establishing inverse polynomial overlap

It is not too hard to establish inverse polynomial overlap $|\alpha_{L+1}(t)|^2 = \Omega(1/\mathrm{polylog}(N))$ for $t = \mathrm{polylog}(N)$. (In fact, it is possible to get perfect overlap as we discuss in the next section.) To prove this result, we use some known results on the spectral properties of $\mathbf{X}'$. Its eigenvectors are

$$|\phi_l\rangle = \sqrt{\frac{2}{L+2}} \sum_{l'=1}^{L+1} \sin\left(\frac{\pi l l'}{L+2}\right) |l'\rangle\ , \tag{C18}$$

and the eigenvalues are

$$\gamma_l = 4 - 2\cos\left(\frac{\pi l}{L+2}\right)\ , \tag{C19}$$

where $l \in [L+1]$. Note that $6 > \gamma_l > 2$. The $|\phi_l\rangle$'s are also the eigenvectors of $\sqrt{\mathbf{X}'}$, whose eigenvalues are $\gamma'_l := \sqrt{\gamma_l}$ and $\sqrt{6} > \gamma'_l > \sqrt{2}$. Let $\Delta_l = \gamma_{l+1} - \gamma_l$ be the spectral gaps of $\mathbf{X}'$ and note that $\Delta_l > 0$ and $\Delta_l/\gamma_l > 0$. The spectral gaps of $\sqrt{\mathbf{X}'}$ satisfy (for $l \in [L]$)

$$\gamma'_{l+1} - \gamma'_l \geq \frac{\pi}{L+2} \min_{x\in(l\pi/(L+2),(l+1)\pi/(L+2))} \frac{d}{dx}\sqrt{4 - 2\cos(x)}$$

$$= \frac{\pi}{L+2} \min_{x\in(l\pi/(L+2),(l+1)\pi/(L+2))} \frac{\sin(x)}{\sqrt{4 - 2\cos(x)}}$$

$$\geq \frac{\pi}{L+2} \min_{x\in(l\pi/(L+2),(l+1)\pi/(L+2))} \frac{\sin(x)}{\sqrt{2}}. \tag{C20}$$

This derivative is zero at $x = 0$ and π, which is outside the region of values of x. The derivative will therefore take its smallest (nonzero) values at the nearest allowed values of x:

$$x = \frac{\pi}{L+2}, \text{ and } x = \frac{\pi(L+1)}{L+2}. \tag{C21}$$

28

We have ($L \geq 1$)

$$\sin\left(\frac{\pi}{L+2}\right) = \sin\left(\frac{\pi(L+1)}{L+2}\right) \geq \frac{1}{\sqrt{2}}\frac{\pi}{L+2}.$$

(C22)

Thus we have

$$\gamma'_{l+1} - \gamma'_l \geq \frac{1}{2}\left(\frac{\pi}{L+2}\right)^2.$$

(C23)

Hence, the smallest spectral gap of $\sqrt{\mathbf{X}'}$, $\Delta' = \min_l(\gamma'_{l+1} - \gamma'_l)$, satisfies $\Delta' = \Omega(1/L^2)$.

The amplitude $\alpha_{L+1}(t)$ that we are interested in can be written as

$$\alpha_{L+1}(t) = \langle L+1| \cos\left(\sqrt{\mathbf{X}'}t\right)|1\rangle$$

$$= \sqrt{\frac{2}{L+2}}\langle L+1|\cos\left(\sqrt{\mathbf{X}'}t\right)\sum_{l=1}^{L+1}\sin\left(\frac{\pi l}{L+2}\right)|\phi_l\rangle$$

$$= \frac{2}{L+2}\sum_{l=1}^{L+1}\cos(\gamma'_l t)\sin\left(\frac{\pi l(L+1)}{L+2}\right)\sin\left(\frac{\pi l}{L+2}\right)$$

$$= \frac{2}{L+2}\sum_{l=1}^{L+1}(-1)^{l-1}\sin^2\left(\frac{\pi l}{L+2}\right)\cos(\gamma'_l t).$$

(C24)

Recall that the probability of measuring $|L+1\rangle$ after evolving with $\sqrt{\mathbf{X}'}$ for time t, when the initial state is $|1\rangle$, is

$$|\alpha_{L+1}(t)|^2 = \left(\frac{2}{L+2}\right)^2\sum_{l,l'=1}^{L+1}(-1)^{l+l'}\sin^2\left(\frac{\pi l}{L+2}\right)\sin^2\left(\frac{\pi l'}{L+2}\right)$$

$$\times \frac{1}{4}\left(e^{it(\gamma'_l+\gamma'_{l'})} + e^{it(\gamma'_l-\gamma'_{l'})} + e^{it(-\gamma'_l+\gamma'_{l'})} + e^{it(-\gamma'_l-\gamma'_{l'})}\right),$$

(C25)

where we used that $\cos(\theta) = \frac{1}{2}(e^{i\theta} + e^{-i\theta})$.

If $l \neq l'$, Eq. (C23) implies $|\gamma'_l - \gamma'_{l'}| \geq \pi^2/(2(L+2)^2) \geq \Delta'$, and $|\gamma'_l + \gamma'_{l'}| \geq 2\sqrt{2}$ for all l, l'. Then, for any $\epsilon > 0$, there exists a probability distribution $f(t)$, where $t \in \{0, 1, \ldots, T\}$ and $T = \mathcal{O}((L+2)^2 \log(1/\epsilon))$, such that $f(t) \geq 0$, $\sum_{t=0}^{T} f(t) = 1$, and

$$\left|\sum_{t=0}^{T}f(t)e^{it(\gamma'_l-\gamma'_{l'})}\right| \leq \epsilon, \qquad \forall\, l \neq l',$$

(C26)

$$\left|\sum_{t=0}^{T}f(t)e^{it(\gamma'_l+\gamma'_{l'})}\right| \leq \epsilon, \qquad \forall\, l, l'.$$

(C27)

In other words, the absolute value of the Fourier transform of $f(t)$ for frequencies ω such that $|\omega| \geq \Delta'$ is upper bounded by ϵ. One choice for $f(t)$ is the probability distribution obtained by taking $m = \mathcal{O}(\log(1/\epsilon))$ samples $\{t_1, \ldots, t_m\}$ from a uniform distribution where $t_i \in \{0, 1, \ldots, T'\}$, $T' = \mathcal{O}(L^2)$, and outputting $t = \sum_i t_i$. Other choices can be found in Ref. [41]. This also implies

$$\left|\sum_{t=0}^{T}f(t)\sum_{l\neq l'}(-1)^{l+l'}\sin^2\left(\frac{\pi l}{L+2}\right)\sin^2\left(\frac{\pi l'}{L+2}\right)e^{it(\gamma'_l-\gamma'_{l'})}\right| \leq \epsilon\sum_{l\neq l'}\sin^2\left(\frac{\pi l}{L+2}\right)\sin^2\left(\frac{\pi l'}{L+2}\right)$$

$$= \epsilon\left(\frac{(L+2)^2}{4} - \frac{3(L+2)}{8}\right)$$

$$\leq \epsilon\frac{(L+2)^2}{4},$$

(C28)

29

and

$$\left| \sum_{t=0}^{T} f(t) \sum_{l,l'} (-1)^{l+l'} \sin^2\left(\frac{\pi l}{L+2}\right) \sin^2\left(\frac{\pi l'}{L+2}\right) e^{it(\gamma_l' + \gamma_{l'}')} \right| \leq \epsilon \sum_{l,l'} \sin^2\left(\frac{\pi l}{L+2}\right) \sin^2\left(\frac{\pi l'}{L+2}\right)$$

$$= \epsilon \frac{(L+2)^2}{4} . \tag{C29}$$

Hence, when we analyze $\sum_{t=0}^{T} f(t) |\alpha_{L+1}(t)|^2$, the terms that dominate the sum are those that correspond to $l = l'$ in Eq. (C25) and where the phases are $e^{\pm it(\gamma_l' - \gamma_{l'}')}$. That is, the above equations give

$$\left| \sum_{t=0}^{T} f(t) |\alpha_{L+1}(t)|^2 - \frac{1}{2} \sum_{t=0}^{T} f(t) \left(\frac{2}{L+2}\right)^2 \sum_{l=1}^{L+1} \sin^4\left(\frac{\pi l}{L+2}\right) \right| \leq 2\left(\frac{2}{L+2}\right)^2 \epsilon \frac{(L+2)^2}{4}$$

$$\leq 2\epsilon . \tag{C30}$$

The second term on the left-hand side can be computed and is $\frac{3}{4(L+2)}$, which gives

$$\left| \sum_{t=0}^{T} f(t) |\alpha_{L+1}(t)|^2 - \frac{3}{4(L+2)} \right| \leq 2\epsilon. \tag{C31}$$

Let us choose $\epsilon = 1/(4(L+2))$. This implies that the average $\sum_{t=0}^{T} f(t) |\alpha_{L+1}(t)|^2$ is $\Omega(1/L)$ and recall that $T = \widetilde{\mathcal{O}}(L^2)$. Then, there must exist a $t = \widetilde{\mathcal{O}}(L^2)$ such that $|\alpha_{L+1}(t)|^2 = \Omega(1/L)$.

Now we are done, since we can run the algorithm for all $t = 1, \ldots, T$, since there are only $\text{poly}(L) = \text{polylog}(N)$ values. Alternatively, we can find the desired t by simulating $\sqrt{\mathbf{X}'}$ for all times $t = 1, 2, \ldots, T$ on a classical computer at cost that is also polynomial in L or $\text{polylog}(N)$. Note that although our proof does not pin down a specific t for which this works, numerical simulations show that a fixed time $t = \mathcal{O}(L)$ suffices for any L.

4. Bonus: Establishing perfect overlap

While the argument above completes the proof of BQP-completeness, we observe that it is also possible to obtain $|\alpha_{L+1}(t)| = 1$ in the proof above, which is equivalent to perfect transmission of a disturbance down a spin chain, which was solved in Refs. [42–44]. The principle is to adjust the weights of the operator $\mathbf{X}'$ as

$$\mathbf{X}' = \sum_{l=0}^{L} b_l |l+1\rangle \langle l+1| - \sum_{l=1}^{L} \sqrt{u_l} \left(|l\rangle \langle l+1| + |l+1\rangle \langle l|\right) , \tag{C32}$$

where we have adjusted the first sum to be consistent with the notation in Ref. [44]. The weights b_l and u_l are adjusted so that the eigenvalues are proportional to the squares of a sequence of integers. Provided the matrix is persymmetric (so $b_l = b_{L-l}$ and $u_l = u_{L+1-l}$), then $e^{i\sqrt{\mathbf{X}'} t} |1\rangle$ gives exactly $|L+1\rangle$ for some t. In particular, in Ref. [42], $\mathbf{X}'$ has the eigenvalues $2k^2$ for $k = 0$ to L, which gives the perfect transfer for $t = \pi/\sqrt{2}$.

Here we cannot use that exact result, because that would imply certain values for the diagonal entries of $\mathbf{X}'$ (and hence $\mathbf{A}$) for which the Hamiltonian would not correspond to a system of coupled oscillators. For our case, it will suffice if the diagonal entries are at least $2\sqrt{2}$ times the off-diagonal entries. To obtain the larger on-diagonal entries we can use the analysis in terms of para-Racah polynominals in [44]. For odd $L = 2j + 1$,

one takes (from Eq. (2.9) of [44], and using $\alpha = 1/2$)

$$b_l = \begin{cases} \frac{1}{2}[a(a+j) + c(c+j) + l(L-l)] & \text{if } l \neq j, j+1 \, , \\ a^2 + \frac{1}{2}j(1+a-c)(1+a+c+j) - \frac{1}{2}(a-c)(1+j)(a+c+j) & \text{if } l = j, j+1 \, , \end{cases} \tag{C33}$$

$$u_l = \begin{cases} \frac{l(L+1-l)(L-l+a+c)(l-1+a+c)((l-j-1)^2 - (a-c)^2)}{4(L-2l)(L-2l+2)} & \text{if } l \neq j+1 \, , \\ \frac{1}{4}(a-c)^2(1+j)^2(a+c+j)^2 & \text{if } l = j+1 \, . \end{cases} \tag{C34}$$

We have flipped the sign of the matrix in Ref. [44] to be consistent with our usage here. The eigenvalues as per Eq. (3.11) of Ref. [44] are (again flipping the sign from that work)

$$\lambda_{2s} = (s+a)^2, \quad s = 0, \ldots, j \, ,$$
$$\lambda_{2s+1} = (s+c)^2, \quad s = 0, \ldots, j \, . \tag{C35}$$

We then get the appropriate set of eigenvalues if $a - c = 1/2$. The matrix is persymmetric if $\alpha = 1/2$. We will also use $L_+ = L + 1$ in the notation to simplify the form of the expressions. Taking $a = L_+/2 + 1/4$ and $c = L_+/2 + 3/4$, we obtain

$$4b_l = 5L_+^2/2 - 1/4 - 2(l - L/2)^2 \, , \tag{C36}$$
$$16u_l = l(2L_+ - l)(L_+^2 - l^2) \, . \tag{C37}$$

Exactly the same result is obtained for odd $L = 2j$ using the expressions in Eq. (4.4) of Ref. [44]. In either case it is found that the eigenvalues are $\frac{1}{4}(L + k + 1/2)^2$ for $k = 0$ to L. Then one can evolve directly to $l = L + 1$ for $t = 2\pi$, so that $|\alpha_{L+1}(2\pi)| = 1$. Alternatively, one can divide the $\mathbf{X}'$ given here by L^2 so the coefficients are $\mathcal{O}(1)$, and evolve for time $2\pi L$.

For the relative values of b_l and u_l, we need to show

$$b_l \geq \sqrt{2}(\sqrt{u_l} + \sqrt{u_{l+1}}) \, , \tag{C38}$$

for $l = 0$ to L. Here we use the convention that $u_0 = 0$, which is given by the formula above. For $l = 0$ we want $b_0 \geq \sqrt{2u_1}$, so Eq. (C38) can still be used with the convention $u_0 = 0$. Note that both b_l and u_l take their maximum values in the centre, but b_l is nonzero at the boundaries whereas u_l is close to zero. This means that $b_l/\sqrt{u_l}$ is smallest for $l = L/2$, and there the ratio is approximately $10/3$, which is significantly larger than $2\sqrt{2}$.

To show this inequality more rigorously, we can use the concavity of u_l as a function of l and the concavity of the square root function to give

$$\sqrt{8u_{l+1/2}} \geq \sqrt{2}(\sqrt{u_l} + \sqrt{u_{l+1}}) \, , \tag{C39}$$

where $u_{l+1/2}$ is using the same function of l (even though it is only meaningful to give coefficients for integer l). We find that

$$b_l^2 - 8u_{l+1/2} = 1 + 16(L_+^4 - L_+^2)[1 + (1-\beta)\beta] + 16L_+^4 \left(3 - \beta\{9 - \beta[5 + 4(2-\beta)\beta]\}\right) \, , \tag{C40}$$

with $\beta = (l + 1/2)/L_+$. Because we consider values of $l \in [0, L]$, we need only consider $\beta \in [0, 1]$. The expression $(3 - \beta\{9 - \beta[5 + 4(2-\beta)\beta]\})$ is positive; it has its minimum of $1/2$ at $\beta = 1/2$. The expression $L_+^4 - L_+^2$ is non-negative, and $1 + (1-\beta)\beta$ is positive for $\beta \in [0, 1]$. As a result

$$b_l^2 \geq 8u_{l+1/2}$$
$$\implies \quad b_l \geq \sqrt{8u_{l+1/2}} \geq \sqrt{2}(\sqrt{u_l} + \sqrt{u_{l+1}}) \, , \tag{C41}$$

as required.

31

Appendix D: Phase estimation approach and proof of Thm. 4

We provide proof of Thm. 4, which provides a method for simulating the dynamics a broader family of classical systems under the harmonic approximation than the previous method described in Thm. 1, at the price of reduced scaling with the desired error tolerance ϵ.

Proof of Thm. 4. We describe the simulation of $\mathbf{H} = -X \otimes \sqrt{\mathbf{A}}$, where X is the single-qubit Pauli bit flip operator and $\sqrt{\mathbf{A}}$ is the principal square root of $\mathbf{A}$, using a standard quantum phase estimation (QPE) approach. In this approach, we run QPE with a unitary that is $\pm$ a walk operator built from a block encoding of $\mathbf{H^{(2)}} := -X \otimes \mathbf{A}$, i.e., a unitary that contains $\mathbf{H^{(2)}}/\Lambda$ in a block, where $\Lambda > 0$ is due to normalization reasons. This QPE provides estimates of the eigenvalues of the walk operator, which can be converted into estimates of the eigenvalues of $\mathbf{H^{(2)}}$ and ultimately of $\mathbf{H}$. Once these estimates are obtained, simulation of $\mathbf{H}$ is simply applying a phase that is a product of the eigenvalue estimate and t. Last, we reverse QPE and other calculations to uncompute the estimated eigenvalues.

In the case where we have oracle access to $\mathbf{K}$, then $\mathbf{A} = \mathbf{BB}^\dagger$ and we can obtain a block encoding of $\mathbf{A}$ – and hence of $\mathbf{H^{(2)}}$ – from the block encodings of $\mathbf{B}$ and $\mathbf{B}^\dagger$. These were given in App. A. This approach would imply $\Lambda = 2\aleph d$. In the case of generalized coordinates described in Sec. VII, where we assume oracle access to d-sparse $\mathbf{A}$ but not $\mathbf{K}$, then $\mathbf{H^{(2)}}$ can be block encoded using standard methods with $\Lambda = \mathcal{O}(\|\mathbf{H^{(2)}}\|_{\max} d) = \mathcal{O}(\|\mathbf{A}\|_{\max} d)$ [24]. Let $|\lambda_j\rangle$ denote the eigenvectors of $\mathbf{A}$ of eigenvalue $\lambda_j \geq 0$. Then, we can write $|\eta_X\rangle |\lambda_j\rangle = |\eta_X, \lambda_j\rangle$ for the eigenvectors of $\mathbf{H^{(2)}}$, where $\eta \in \{0,1\}$, and $|0_X\rangle = |-\rangle$ and $|1_X\rangle = |+\rangle$ are the eigenvectors of X. The corresponding eigenvalues of $\mathbf{H^{(2)}}$ are $\gamma_{\eta,j} := (-1)^\eta \lambda_j$. Our first step is to estimate the eigenvalue $\gamma_{\eta,j}$ within fixed error, and then propagate that error into the maximum error that can be observed in $(-1)^\eta \sqrt{\lambda_j}$, which is the corresponding eigenvalue of $\mathbf{H}$, using an arithmetic circuit on the outputs of the QPE routine. Phase estimation can be used to provide a confidence interval $S_{\eta,j}$ for the eigenvalue estimates of $\mathbf{H^{(2)}}$, which we call x. That is, for all $x \in S_{\eta,j}$ that are estimates of $\gamma_{\eta,j}$, and for a given ϵ_{PE}, we define $\epsilon_{\eta,j}(x) := x - \gamma_{\eta,j}$ and $S_{\eta,j} := \{x : \epsilon_{\eta,j}(x) \in [-\epsilon_{\mathrm{PE}}, \epsilon_{\mathrm{PE}}]\}$. To describe the contribution to estimates outside the confidence interval we use a state $|\phi_{\eta,j}\rangle$ of unit norm and an amplitude $\sqrt{\delta_{\eta,j}}$, where $\delta_{\eta,j} > 0$ depends on η and j, and $1 - \delta_{\eta,j}$ is the confidence level when the input state is $|\eta_X, \lambda_j\rangle$. Eventually, we will set $\delta_{\eta,j} \leq \delta_{\mathrm{PE}}$ for all η and j, where $\delta_{\mathrm{PE}} > 0$ is determined below. Then, when implementing QPE on input eigenstate $|\eta_X, \lambda_j\rangle$ and using a unitary that provides eigenvalue estimates of $\mathbf{H^{(2)}}$ (e.g., a walk operator), the state is approximately transformed as

$$|\eta_X, \lambda_j\rangle \mapsto |\eta_X, \lambda_j\rangle \left(\sqrt{1-\delta_{\eta,j}} \sum_{x \in S_{\eta,j}} b_x |x\rangle + \sqrt{\delta_{\eta,j}} |\phi_{\eta,j}\rangle \right), \tag{D1}$$

for some unit vector $\vec{b}$ that gives the probability distribution $|b_x|^2$ for phase estimates within the confidence interval (the states $|x\rangle$ can be basis states that encode the estimates of $\gamma_{\eta,j}$). When performing QPE using Kaiser windows [34], the number of invocations of the walk operator (i.e., the query complexity) built from the block encoding of $\mathbf{H^{(2)}}$ is then [45]

$$\mathcal{O}(\Lambda \log(1/\delta_{\mathrm{PE}})/\epsilon_{\mathrm{PE}}) = \mathcal{O}(\|\mathbf{A}\|_{\max} d \log(1/\delta_{\mathrm{PE}})/\epsilon_{\mathrm{PE}}). \tag{D2}$$

The walk operator we use in QPE combines the block encoding of $\mathbf{H^{(2)}}$ with other two-qubit gates, including a reflection on some ancilla qubits [46]. That is, each use of the walk operator requires $\mathcal{O}(1)$ uses of the oracle to access $\mathbf{A}$. The eigenvalues of the walk operator are then $\mp e^{\pm i \arcsin(\lambda_j/\Lambda)}$. The actual eigenvalues of $\mathbf{H^{(2)}}$, which are $\gamma_{\eta,j}$, can be estimated by QPE using the walk operator, then taking the sine of the result to give an estimate of $\gamma_{\eta,j}/\Lambda$. Note that $|\gamma_{\eta,j}|/\Lambda \leq 1$ for all $\eta \in \{0,1\}$ and $j \in [N]$. Aiming for $\epsilon_{\mathrm{PE}}/\Lambda$ uncertainty in the phase estimation of $\arcsin(\lambda_j/\Lambda)$ yields the following estimate in $\gamma_{\eta,j}/\Lambda$:

$$|\gamma_{\eta,j}/\Lambda - \sin(\arcsin(\gamma_{\eta,j}/\Lambda) + \epsilon_{\mathrm{PE}}/\Lambda)| \leq \epsilon_{\mathrm{PE}}/\Lambda. \tag{D3}$$

Then, in QPE we are first obtaining an estimate $x = \gamma_{\eta,j} + \epsilon_{\eta,j}(x)$, where the error $\epsilon_{\eta,j}(x)$ satisfies $|\epsilon_{\eta,j}(x)| \leq \epsilon_{\mathrm{PE}}$ within a $(1 - \delta_{\mathrm{PE}})$-confidence level. We want to convert this estimate into an estimate of

an eigenvalue of $\mathbf{H}$, i.e., an estimate of $(-1)^\eta \sqrt{\lambda_j} = \operatorname{sign}(\gamma_{\eta,j})\sqrt{|\gamma_{\eta,j}|}$, which is obtained by computing $\operatorname{sign}(x)\sqrt{|x|}$. This gives the error

$$
\begin{aligned}
\Delta_{\eta,j}(x) &:= \operatorname{sign}(x)\sqrt{|x|} - \operatorname{sign}(\gamma_{\eta,j})\sqrt{|\gamma_{\eta,j}|} \\
&= \operatorname{sign}(\gamma_{\eta,j} + \epsilon_{\eta,j}(x))\sqrt{|\gamma_{\eta,j} + \epsilon_{\eta,j}(x)|} - \operatorname{sign}(\gamma_{\eta,j})\sqrt{|\gamma_{\eta,j}|},
\end{aligned}
\tag{D4}
$$

which we seek to bound. In the case where $\gamma_{\eta,j} \geq 0$ (i.e., $\eta = 0$ and $\gamma_{\eta,j} = \lambda_j$), we have

$$
|\Delta_{\eta,j}(x)| = \left| \operatorname{sign}(\lambda_j + \epsilon_{\eta,j}(x))\sqrt{|\lambda_j + \epsilon_{\eta,j}(x)|} - \sqrt{\lambda_j} \right|,
\tag{D5}
$$

and in the case where $\gamma_{\eta,j} \leq 0$ (i.e., $\eta = 1$ and $\gamma_{\eta,j} = -\lambda_j$), we have

$$
|\Delta_{\eta,j}(x)| = \left| \operatorname{sign}(-\lambda_j + \epsilon_{\eta,j}(x))\sqrt{|-\lambda_j + \epsilon_{\eta,j}(x)|} + \sqrt{\lambda_j} \right|.
\tag{D6}
$$

If we replace $\epsilon_{\eta,j}(x)$ with $-\epsilon_{\eta,j}(x)$ this becomes

$$
\left| \operatorname{sign}(\lambda_j + \epsilon_{\eta,j}(x))\sqrt{|\lambda_j + \epsilon_{\eta,j}(x)|} - \sqrt{\lambda_j} \right|,
\tag{D7}
$$

and so bounding the expression for the case $\eta = 0$ will be sufficient to account for all cases. To find the bound for this case we will show that for any $a \in \mathbb{R}$

$$
\left| \operatorname{sign}(1 + a)\sqrt{|1 + a|} - 1 \right| \leq \sqrt{2}\min(|a|, \sqrt{|a|}).
\tag{D8}
$$

There are four cases to consider to prove this expression.

1. For $a \geq 1$ we can prove the inequality by using

$$
\begin{aligned}
1 + a \leq 1 + a + 2\sqrt{a} &\implies \sqrt{1 + a} \leq 1 + \sqrt{a} \\
&\implies \sqrt{1 + a} - 1 \leq \sqrt{a} \\
&\implies \operatorname{sign}(1 + a)\sqrt{|1 + a|} - 1 \leq \sqrt{|a|}.
\end{aligned}
\tag{D9}
$$

Since $\sqrt{|a|} \leq |a|$ for $a \geq 1$, this proves the inequality.

2. For $a \in [0, 1]$, we can use

$$
\begin{aligned}
\sqrt{1 + a} \leq 1 + a &\implies \sqrt{|1 + a|} - 1 \leq |a| \\
&\implies \operatorname{sign}(1 + a)\sqrt{|1 + a|} - 1 \leq |a|.
\end{aligned}
\tag{D10}
$$

Since $|a| \leq \sqrt{|a|}$ for $a \in [0, 1]$, this proves the inequality.

3. For $a \in [-1, 0]$,

$$
\begin{aligned}
\sqrt{|1 + a|} \geq |1 + a| &\implies \sqrt{|1 + a|} - 1 \geq a \\
&\implies 1 - \sqrt{|1 + a|} \leq -a \\
&\implies \left| \operatorname{sign}(1 + a)\sqrt{|1 + a|} - 1 \right| \leq |a|.
\end{aligned}
\tag{D11}
$$

Again for $a \in [-1, 0]$ we can use $|a| \leq \sqrt{|a|}$. This time we need to take the absolute value because $\sqrt{|1 + a|} \leq 1$.

4. For $a \leq -1$ we can use

$$
\begin{aligned}
0 \leq (|a| - 2)^2 &\implies 4(|a| - 1) \leq a^2 \\
&\implies 2\sqrt{|a| - 1} \leq |a| \\
&\implies (|a| - 1) + 2\sqrt{|a| - 1} + 1 \leq 2|a| \\
&\implies \sqrt{|a| - 1} + 1 \leq \sqrt{2|a|} \\
&\implies \left| \mathrm{sign}(1 + a)\sqrt{|1 + a|} - 1 \right| \leq \sqrt{2|a|} .
\end{aligned}
\tag{D12}
$$

That also implies that it is upper bounded by $\sqrt{2}|a|$.

The result gives us the bound on the error in the signed square root as follows:

$$
\Delta_{\eta,j} := \max_{x:|x - \gamma_{\eta,j}| \leq \epsilon_{\mathrm{PE}}} |\Delta_{\eta,j}(x)| \leq \min\left(\epsilon_{\mathrm{PE}}\sqrt{2/\lambda_j}, \sqrt{2\epsilon_{\mathrm{PE}}} \right) .
\tag{D13}
$$

Hence, from the estimate x of $\gamma_{\eta,j}$ within error at most ϵ_{PE}, we can compute an estimate of $(-1)^\eta\sqrt{\lambda_j}$, the eigenvalue of $\mathbf{H}$, within error at most $\Delta_{\eta,j}$ as above.

To implement $e^{itX \otimes \sqrt{\mathbf{A}}}$, we apply a phase factor proportional to the estimated eigenvalue from QPE. Then the state after applying the phase factor is transformed as

$$
|\eta_X, \lambda_j\rangle \mapsto |\eta_X, \lambda_j\rangle \left(\sqrt{1 - \delta_{\eta,j}} \sum_{x \in S_{\eta,j}} b_x e^{-it\,\mathrm{sign}(x)\sqrt{|x|}} |x\rangle + \sqrt{\delta_{\eta,j}}|\phi'_{\eta,j}\rangle \right)
$$

$$
= e^{-it(-1)^\eta\sqrt{\lambda_j}} |\eta_X, \lambda_j\rangle \left(\sqrt{1 - \delta_{\eta,j}} \sum_{x \in S_{\eta,j}} b_x e^{-it\Delta_{\eta,j}(x)} |x\rangle + \sqrt{\delta_{\eta,j}}|\phi'_{\eta,j}\rangle \right)
$$

$$
= e^{-it(-1)^\eta\sqrt{\lambda_j}} |\eta_X, \lambda_j\rangle \left[\left(\sqrt{1 - \delta_{\eta,j}} \sum_{x \in S_{\eta,j}} b_x |x\rangle + \sqrt{\delta_{\eta,j}} |\phi_{\eta,j}\rangle \right) \right.
$$

$$
\left. + \left(\sqrt{1 - \delta_{\eta,j}} \sum_{x \in S_{\eta,j}} b_x (e^{-it\Delta_{\eta,j}(x)} - 1) |x\rangle + \sqrt{\delta_{\eta,j}}(|\phi'_{\eta,j}\rangle - |\phi_{\eta,j}\rangle) \right) \right] .
\tag{D14}
$$

The states $|\phi_{\eta,j}\rangle$ and $|\phi'_{\eta,j}\rangle$ are states of the ancillas of unit norm, supported in the space orthogonal to $\{|x\rangle : x \in S_{\eta,j}\}$. After the phase was applied, we need to invert QPE. The inverse phase estimation and projection onto the zero state of the ancillas that were used to estimate the eigenvalues corresponds to applying

$$
\langle\tilde{\lambda}_{\eta,j}| = \sum_{x \in S_{\eta,j}} b_x^* \langle x| ,
\tag{D15}
$$

to give

$$
e^{-it(-1)^\eta\sqrt{\lambda_j}} |\eta_X, \lambda_j\rangle \left[\sqrt{1 - \delta_{\eta,j}} + \sqrt{\delta_{\eta,j}}\langle\tilde{\lambda}_{\eta,j}|\phi_{\eta,j}\rangle \right.
$$

$$
\left. + \sqrt{1 - \delta_{\eta,j}} \sum_{x \in S_{\eta,j}} |b_x|^2(e^{-it\Delta_{\eta,j}(x)} - 1) + \sqrt{\delta_{\eta,j}}\left(\langle\tilde{\lambda}_{\eta,j}|\phi'_{\eta,j}\rangle - \langle\tilde{\lambda}_{\eta,j}|\phi_{\eta,j}\rangle \right) \right] .
\tag{D16}
$$

The error can be bounded from the Euclidean distance between the above state and the correct state $e^{-it(-1)^\eta\sqrt{\lambda_j}} |\eta_X, \lambda_j\rangle$. Using $\delta_{\eta,j} \leq \delta_{\mathrm{PE}}$, this distance is upper bounded by

$$
1 - \sqrt{1 - \delta_{\mathrm{PE}}} + 3\sqrt{\delta_{\mathrm{PE}}} + \sqrt{1 - \delta_{\mathrm{PE}}}\langle e^{-it\Delta_{\eta,j}(x)} - 1\rangle ,
\tag{D17}
$$

34

where the expectation value is on the probability distribution given by $|b_x|^2$. This can be further upper bounded by

$$\max_{\eta,j} \Delta_{\eta,j} t + 4\sqrt{\delta_{\text{PE}}} \,. \tag{D18}$$

To appropriately bound the error by ϵ as required by Problem 4, we can take

$$4\sqrt{\delta_{\text{PE}}} = \epsilon/2 \,, \tag{D19}$$

so $\delta_{\text{PE}} = \epsilon^2/64$. Because measurement of the phase with this confidence level has a factor $\mathcal{O}(\log(1/\delta_{\text{PE}}))$ in the complexity in Eq. (D2), it corresponds to a factor of $\mathcal{O}(\log(1/\epsilon))$. Then for

$$\max_{\eta,j} \Delta_{\eta,j} \leq \max_{\eta,j} \min\left(\epsilon_{\text{PE}}\sqrt{2/\lambda_j}, \sqrt{2\epsilon_{\text{PE}}}\right)$$
$$= \min\left(\epsilon_{\text{PE}}\sqrt{2/\min_j \lambda_j}, \sqrt{2\epsilon_{\text{PE}}}\right), \tag{D20}$$

to satisfy $\max_{\eta,j} \Delta_{\eta,j} t \leq \epsilon/2$, we can take

$$\epsilon_{\text{PE}} = \max\left(\frac{\epsilon\sqrt{\min_j \lambda_j}}{2t\sqrt{2}}, \frac{\epsilon^2}{8t^2}\right). \tag{D21}$$

Using these expressions in Eq. (D2) gives us an overall query complexity

$$\mathcal{O}\left(\|\mathbf{A}\|_{\max} d \log(1/\epsilon) \min\left(\frac{t\sqrt{\|\mathbf{A}^{-1}\|}}{\epsilon}, \frac{t^2}{\epsilon^2}\right)\right) \tag{D22}$$

for the phase estimation approach. This is the stated result.

There are further elementary two-qubit gates arising from three main areas.

1. Computing $\text{sign}(x)\sqrt{|x|}$, the estimate of the eigenvalue of $\mathbf{H}$, multiplying by t, and then applying the phase factor. The dominant complexity is that from computing the square root, which scales using Newton iteration and textbook multiplication as $\widetilde{\mathcal{O}}(\log^2(1/\epsilon))$. The phase rotation then requires a linear number of controlled rotation gates based on the target, which in turn requires at most $\mathcal{O}(\log(1/\epsilon))$ two-qubit gates to implement. Thus the former cost dominates.

2. The gates for the implementation of the block encoding will be logarithmic in N and the allowable error of the block encoding. Because the allowable error in the block encoding needs to be ϵ divided by the number of block encodings in the phase estimation, there will be logarithmic factors in many of the parameters here.

3. The gate complexity of preparing the control states states for phase estimation with optimal confidence intervals is at most linear in the dimension of this control register [47], which is the same as the number of oracle calls. This is the least significant contribution to the complexity and gives no logarithmic factors.

In quoting the complexity we give the logarithmic factor in N coming from implementing the block encoding, but for simplicity use $\widetilde{\mathcal{O}}$ and do not explicitly give the logarithmic factors in other parameters. $\quad\square$

Appendix E: Initial state preparation

Our algorithm for solving Problem 1 accepts as input a circuit $\mathcal{W}$ that prepares an initial state of the form in Eq. (2) for $t = 0$. In this appendix we explain how we might create such a circuit given only the ability to

separately create superpositions over the initial positions and initial velocities. Our construction is related to that in Lemma 8 of App. A.

As in Sec. IV, we consider a setting where the $N \times N$ matrices $\mathbf{M}$ and $\mathbf{K}$ can be queried through the use of a unitary $\mathcal{S}$ that gives the positions and non-zero entries of these matrices.

Lemma 10. *Let $\mathbf{K}$ be the $N \times N$ symmetric and d-sparse matrix of spring constants, $\kappa_{\max} \geq \kappa_{jk} \geq 0$, $\mathbf{M} \succ 0$ be the $N \times N$ diagonal matrix of masses, and $m_{\max} \geq m_j > 0$, where $\kappa_{\max}$ and $m_{\max}$ are known. Assume we are given access to $\mathbf{K}$ and $\mathbf{M}$ through an oracle $\mathcal{S}$, and access to a unitary $\mathcal{U}$ that performs the map*

$$\alpha \ket{0} \mapsto \ket{0}\ket{\dot{\vec{x}}(0)}, \qquad \beta \ket{1} \mapsto \ket{1}\ket{\vec{x}(0)},$$

where

$$\ket{\dot{\vec{x}}(0)} = \sum_{j=1}^{N} \dot{x}_j(0)\ket{j}, \qquad \ket{\vec{x}(0)} = \sum_{j=1}^{N} x_j(0)\ket{j},$$

are normalized states that encode the initial states of the oscillators in their amplitudes, and α and β are known norms of the vectors $\dot{\vec{x}}(0)$ and $\vec{x}(0)$, respectively. Then, there exists a quantum algorithm that prepares a state that is ϵ-close in Euclidian norm to

$$\ket{\psi(0)} = \frac{1}{\sqrt{2E}} \begin{pmatrix} \sqrt{\mathbf{M}}\dot{\vec{x}}(0) \\ i\vec{\mu}(0) \end{pmatrix}, \tag{E1}$$

where $E > 0$ is a known constant (the energy of the classical oscillators) and $\vec{\mu}(0) \in \mathbb{R}^M$ for $M = N(N+1)/2$ is a vector whose entries are $\sqrt{\kappa_{jj}}x_j(0)$ or $\sqrt{\kappa_{jk}}(x_j(0) - x_k(0))$, $k > j$. The quantum circuit makes $Q_{\mathrm{ini}} = \mathcal{O}(\sqrt{E_{\max}d/E})$ uses of $\mathcal{U}$, $\mathcal{S}$, and its inverses, in addition to

$$G_{\mathrm{ini}} = \mathcal{O}(\sqrt{E_{\max}d/E}\, \mathrm{polylog}(NE_{\max}/(E\epsilon))) \tag{E2}$$

two-qubit gates. Here $E_{\max} = \frac{m_{\max}}{2}\sum_j (\dot{x}_j(0))^2 + \frac{\kappa_{\max}}{2}\sum_j (x_j(0))^2$ is the energy of a system of N uncoupled oscillators where all masses are $m_{\max}$ and all individual spring constants are $\kappa_{\max}$, and for the same initial conditions.

Proof. For ease of implementation, we can encode the state on $2n + 1$ qubits, where $n = \log(N)$. In this space, the initial state is represented as

$$\ket{\psi(0)} = \frac{1}{\sqrt{2E}}\left[\sum_j \sqrt{m_j}\dot{x}_j(0)\ket{j}\ket{0} + i\sum_j \sqrt{\kappa_{jj}}x_j(0)\ket{j}\ket{j} + i\sum_{j<k}(\sqrt{\kappa_{jk}}(x_j(0) - x_k(0))\ket{j}\ket{k}\right]. \tag{E3}$$

First we rotate a qubit and apply the state preparation oracle as

$$\ket{0} \mapsto \frac{1}{\sqrt{m_{\max}\alpha^2 + 2\kappa_{\max}d\,\beta^2}}(\sqrt{m_{\max}}\alpha\ket{0} + i\sqrt{2\kappa_{\max}d}\beta\ket{1})$$

$$\mapsto \frac{1}{\sqrt{m_{\max}\alpha^2 + 2\kappa_{\max}d\,\beta^2}}(\sqrt{m_{\max}}\alpha\ket{0}\ket{\dot{\vec{x}}(0)} + i\sqrt{2\kappa_{\max}d}\beta\ket{1}\ket{\vec{x}(0)}). \tag{E4}$$

The way we have described the oracle for $\dot{\vec{x}}(0)$ and $\vec{x}(0)$ allows for the case where α or β may be zero, in which case $\mathcal{U}$ can be arbitrary on that subspace because it has no effect on the state above. Then our goal is to apply $\sqrt{\mathbf{M}}$ to the $\ket{\dot{\vec{x}}(0)}$ portion, and $\mathbf{B}^\dagger\sqrt{\mathbf{M}}$ to the $\ket{\vec{x}(0)}$ portion.

The most challenging part is for applying $\mathbf{B}^\dagger\sqrt{\mathbf{M}}$. To do this, we essentially apply the same construction in Lemma 8 in App. A, but for block encoding $\mathbf{B}^\dagger\sqrt{\mathbf{M}}$ rather than $\mathbf{B}^\dagger$. First consider the operation where we prepare a superposition over d values in an ancilla to give

$$\ket{\vec{x}(0)}\frac{1}{\sqrt{d}}\sum_{\ell=1}^{d}\ket{\ell}. \tag{E5}$$

This preparation is simple if d is a power of 2, and otherwise can be performed using $\mathcal{O}(\log d) = \mathcal{O}(n)$ gates via amplitude amplification (see App. E.2 of [34]). We can then apply the oracle for the positions of non-zero entries of $\mathbf{K}$ to map k to the non-zero entry in row j, and obtain

$$\frac{1}{\sqrt{d}} \sum_{j=1}^{N} x_j(0) |j\rangle \sum_{k \in [N]:\kappa_{jk} \neq 0} |k\rangle. \tag{E6}$$

There may be less than d values of k for each j such that $\kappa_{jk} \neq 0$, but the oracle may be chosen to give dummy values of k to pad it out to d. These will later be eliminated because κ_{jk} is actually zero for those values of j, k.

In the usual way, we can use the oracle for the matrix entries of κ_{jk} to output its value, and perform an inequality test with an ancilla in an equal superposition to apply a factor corresponding to $\sqrt{\kappa_{jk}/\kappa_{\max}}$ (which is no greater than 1) [38]. This will give us, with an amplitude corresponding to the amplitude for success,

$$\frac{1}{\sqrt{\kappa_{\max}d}} \sum_{j,k} \sqrt{\kappa_{jk}} \, x_j(0) |j\rangle |k\rangle. \tag{E7}$$

We then perform an inequality test, and perform a controlled swap based on the result of the inequality test. This gives the state

$$\frac{1}{\sqrt{\kappa_{\max}d}} \left[\sum_{j=1}^{N} \sqrt{\kappa_{jj}} \, x_j(0) |j\rangle |j\rangle |0\rangle + \sum_{k>j} \sqrt{\kappa_{jk}} \, x_j(0) |j\rangle |k\rangle |0\rangle + \sum_{k<j} \sqrt{\kappa_{jk}} \, x_j(0) |j\rangle |k\rangle |1\rangle \right], \tag{E8}$$

where the third register is the qubit flagging the result of the inequality test. Then we perform a Z gate on that qubit and relabel the third sum to give

$$\frac{1}{\sqrt{\kappa_{\max}d}} \left[\sum_{j=1}^{N} \sqrt{\kappa_{jj}} \, x_j(0) |j\rangle |j\rangle |0\rangle + \sum_{j<k} \sqrt{\kappa_{jk}} \, |j\rangle |k\rangle \left[x_j(0) |0\rangle - x_k(0) |1\rangle \right] \right]. \tag{E9}$$

Projecting onto the $|+\rangle$ state on the ancilla qubit then gives

$$\frac{1}{\sqrt{2\kappa_{\max}d}} \left[\sum_{j=1}^{N} \sqrt{\kappa_{jj}} \, x_j(0) |j\rangle |j\rangle + \sum_{j<k} \sqrt{\kappa_{jk}} \left[x_j(0) - x_k(0) \right] |j\rangle |k\rangle \right], \tag{E10}$$

where the amplitude is indicating the amplitude for success.

To apply $\sqrt{\mathbf{M}}$ to the $|\dot{\vec{x}}(0)\rangle$ portion, we can simply perform the same inequality testing procedure from [38] to apply a factor of $\sqrt{m_j/m_{\max}}$, and give

$$|\dot{\vec{x}}(0)\rangle \mapsto \frac{1}{\sqrt{m_{\max}}} \sum_{j=1}^{N} \sqrt{m_j} \dot{x}_j(0) |j\rangle. \tag{E11}$$

Combined, the preparation on these two parts of the state gives

$$\frac{1}{\sqrt{m_{\max}\alpha^2 + 2\kappa_{\max}d\,\beta^2}} \left(\sqrt{m_{\max}}\alpha |0\rangle |\dot{\vec{x}}(0)\rangle + i2\sqrt{\kappa_{\max}d}\beta |1\rangle |\vec{x}(0)\rangle \right)$$

$$\mapsto \frac{1}{\sqrt{m_{\max}\alpha^2 + 2\kappa_{\max}d\,\beta^2}} \left[\sum_{j=1}^{N} \sqrt{m_j} \, \dot{x}_j(0) |0\rangle |j\rangle |0\rangle \right.$$

$$\left. + i \sum_{j=1}^{N} \sqrt{\kappa_{jj}} \, x_j(0) |1\rangle |j\rangle |j\rangle + i \sum_{j<k} \sqrt{\kappa_{jk}} \, (x_j(0) - x_k(0)) |1\rangle |j\rangle |k\rangle \right], \tag{E12}$$

which is the correct state (Eq. (E3)), but subnormalised indicating that it is not produced deterministically and we need to use amplitude amplification.

The number of amplitude amplification rounds scales as the inverse of the amplitude, so the complexity in terms of α, β, E is

$$\mathcal{O}\left(\sqrt{\frac{m_{\max}\alpha^2 + 2\kappa_{\max}d\,\beta^2}{2E}}\right). \tag{E13}$$

We assume we know the constants for this Lemma to simplify the amplitude amplification. It is also possible to perform amplitude amplification when the amplitude is unknown but bounded [48]. Note that

$$\frac{1}{2}m_{\max}\alpha^2 = \frac{1}{2}\sum_{j=1}^{N}m_{\max}(\dot{x}_j(0))^2 \tag{E14}$$

$$=: K_{\max}, \tag{E15}$$

which is the kinetic energy of the system of oscillators at $t = 0$ if all masses were $m_j = m_{\max}$. Also,

$$\frac{1}{2}\kappa_{\max}\beta^2 = \frac{1}{2}\sum_{j=1}^{N}\kappa_{\max}(x_j(0))^2 \tag{E16}$$

$$=: U_{\max} \tag{E17}$$

is the potential energy of a system of N uncoupled oscillators at $t = 0$ if all spring constants were $\kappa_{jj} = \kappa_{\max}$ and $\kappa_{jk} = 0$ if $j \neq k$. If $E_{\max} = K_{\max} + U_{\max} = \frac{m_{\max}}{2}\|\dot{\vec{x}}(0))\|^2 + \frac{\kappa_{\max}}{2}\|\vec{x}(0)\|^2$ is the total energy of such a system, it satisfies

$$m_{\max}\alpha^2 + 2\kappa_{\max}d\,\beta^2 \leq 4E_{\max}d. \tag{E18}$$

The number of amplitude amplification rounds is then $\mathcal{O}(\sqrt{E_{\max}d/E})$. Each amplitude amplification round makes two uses of $\mathcal{S}, \mathcal{S}^\dagger$, and also one use of $\mathcal{U}$ and $\mathcal{U}^\dagger$. Hence, the overall query complexity of our algorithm that prepares $|\psi(0)\rangle$ is

$$Q_{\mathrm{ini}} = \mathcal{O}\left(\sqrt{\frac{E_{\max}d}{E}}\right). \tag{E19}$$

To find the accuracy of the block encoding we need to account for the error due to applying the factors of $\sqrt{\kappa_{jk}/\kappa_{\max}}$ and $\sqrt{m_j/m_{\max}}$ by inequality testing. If we give the maximum errors in these factors as δ

(corresponding to $\log(1/\delta)$ bits in the inequality testing), then the error in Eq. (E12) is upper bounded as

$$\frac{1}{\sqrt{m_{\max}\alpha^2 + 2\kappa_{\max}d\,\beta^2}}\left\|\sum_j \delta\,\sqrt{m_{\max}}\dot{x}_j(0)\,|0\rangle\,|j\rangle\,|0\rangle + \sum_j \delta\,\sqrt{\kappa_{\max}}x_j(0)\,|1\rangle\,|j\rangle\,|j\rangle\right.$$

$$\left. + \sum_{\substack{j<k \\ \kappa_{jk}\neq 0}} \delta\,\sqrt{\kappa_{\max}}(x_j(0) - x_k(0))\,|1\rangle\,|j\rangle\,|k\rangle\right\|$$

$$\leq \frac{1}{\sqrt{m_{\max}\alpha^2 + 2\kappa_{\max}d\,\beta^2}}\left[m_{\max}\left\|\sum_{j=1}^N \delta\,\dot{x}_j(0)\,|j\rangle\right\|^2 + \kappa_{\max}\left\|\sum_{\substack{j\leq k \\ \kappa_{jk}\neq 0}} \delta\,x_j(0)\,|j\rangle\,|k\rangle - \sum_{\substack{j>k \\ \kappa_{jk}\neq 0}} \delta\,x_j(0)\,|k\rangle\,|j\rangle\right\|^2\right]^{1/2}$$

$$\leq \frac{\delta}{\sqrt{m_{\max}\alpha^2 + 2\kappa_{\max}d\,\beta^2}}\left[m_{\max}\left\|\dot{\vec{x}}_j(0)\right\|^2 + \kappa_{\max}\left\|\sqrt{2}\sum_{j,k;\kappa_{jk}\neq 0} x_j(0)\,|j\rangle\,|k\rangle\right\|^2\right]^{1/2}$$

$$\leq \frac{\delta}{\sqrt{m_{\max}\alpha^2 + 2\kappa_{\max}d\,\beta^2}}\left[m_{\max}\left\|\dot{\vec{x}}(0)\right\|^2 + 2\kappa_{\max}d\,\|\vec{x}(0)\|^2\right]^{1/2}$$

$$= \delta\,. \tag{E20}$$

The inequality on the third line is obtained by noting that we can move from the state on the third line to that on the second line by the procedure described above, with an inequality test between j and k, a controlled swap and phase, then projection onto $|+\rangle$ on the ancilla. Thus, δ corresponds to the error *before* amplitude amplification, and it is amplified by a factor of $\mathcal{O}(\sqrt{E_{\max}d/E})$.

If we are aiming for error ϵ in the state preparation, we should therefore take

$$\delta = \mathcal{O}\left(\epsilon\sqrt{\frac{E}{E_{\max}d}}\right). \tag{E21}$$

Because the state preparation by inequality testing uses squares, we have a gate complexity that is the square of $\log(1/\delta)$ for each step of amplitude amplification, and therefore the gate complexity from this source is

$$\mathcal{O}\left(\sqrt{\frac{E_{\max}d}{E}}\log^2\left(\frac{E_{\max}d}{E\epsilon}\right)\right). \tag{E22}$$

Moreover, there are $\mathcal{O}(\log(N))$ gates needed to implement an inequality test and controlled swap for each step of amplitude amplification. Using $d \leq N$, we can give the overall gate complexity as

$$\mathcal{O}\left(\sqrt{\frac{E_{\max}d}{E}}\log^2\left(\frac{E_{\max}N}{E\epsilon}\right)\right), \tag{E23}$$

which are the stated results. $\qquad\square$

The factor $E_{\max}/E$ in the complexity can become large when the masses m_j or spring constants κ_{jk} lack uniformity. In this case, the state $|\psi(0)\rangle$ is more complicated, and one has to "pay extra" for its preparation. In addition, the query complexity in Lem. 10 matches a lower bound for some instances. For example, consider the case where $x_j(0) = 0$ for all $j \in [N]$, which implies $\beta^2 = 0$, and $\dot{x}_j(0) = 1/\sqrt{N}$ for all $j \in [N]$, which implies $\alpha^2 = 1$. Let the masses be $m_j = 1$, for some unknown j, and $m_{j'} = 0$ otherwise. Then, $|\psi(0)\rangle$ is simply a state $|j\rangle$ in a computational basis, and our state preparation algorithm would output the unknown j with high probability, as in the unstructured search problem [49]. A lower bound on the query complexity for this case is $\Omega(\sqrt{N})$ [50]. Since $E_{\max} = \frac{1}{2}m_{\max}\alpha^2 = \frac{1}{2}$, $E = \frac{1}{2}(\dot{x}_j(0))^2 = \frac{1}{2N}$, and $d = 1$ for this instance, the query complexity of our approach is $\mathcal{O}(\sqrt{N})$, matching the lower bound.

Appendix F: Other encodings

In Sec. III, we choose a specific encoding for the position and velocities as a quantum state $|\psi\rangle$ in Eq. (14), which satisfies Eq. (13). Let $|\psi(t)\rangle \in \mathbb{C}^{N+M}$ be written as

$$|\psi(t)\rangle = \begin{pmatrix} \vec{\nu}(t) \\ i\vec{\mu}(t) \end{pmatrix}, \tag{F1}$$

where $\vec{\nu}(t) \in \mathbb{C}^N$ and $i\vec{\mu}(t) \in \mathbb{C}^M$. We made the choice of $\vec{\nu}(t) = \dot{\vec{y}}(t)$ and $\vec{\mu}(t) = \mathbf{B}^\dagger \vec{y}(t)$ to present our main results. Nevertheless, other choices also work, and we now discuss another solution (and problem). This provides us with a new encoding that is different from the one we used in Problem 1, and is closely related to the encoding used in Ref. [11].

Let $\vec{\nu}(t) = \mathbf{P}\vec{y}(t)$, where $\mathbf{P}$ projects out the components of $\vec{y}(t)$ corresponding to the null space of $\mathbf{A}$ (or $\mathbf{B}^\dagger$). Let $\vec{\mu}(t) = -\mathbf{B}^+\mathbf{P}\dot{\vec{y}}(t)$, where $\mathbf{B}^+$ is the Moore-Penrose pseudo-inverse of $\mathbf{B}$. This is an $M \times N$ matrix that satisfies $\mathbf{B}\mathbf{B}^+\mathbf{B} = \mathbf{B}$ and $\mathbf{B}^+ = \mathbf{B}^\dagger(\mathbf{B}\mathbf{B}^\dagger)^+ = \mathbf{B}^\dagger\mathbf{A}^+$. We claim that this choice also satisfies Eq. (13). Specifically, we want it to satisfy

$$\begin{pmatrix} \dot{\vec{\nu}}(t) \\ i\dot{\vec{\mu}}(t) \end{pmatrix} = -i\mathbf{H} \begin{pmatrix} \vec{\nu}(t) \\ i\vec{\mu}(t) \end{pmatrix} = \begin{pmatrix} -\mathbf{B}\vec{\mu}(t) \\ i\mathbf{B}^\dagger\vec{\nu}(t) \end{pmatrix}. \tag{F2}$$

We then have

$$\dot{\vec{\nu}}(t) = \mathbf{P}\dot{\vec{y}}(t) = \mathbf{A}\mathbf{A}^+\mathbf{P}\dot{\vec{y}}(t) = \mathbf{B}\mathbf{B}^+\mathbf{P}\dot{\vec{y}}(t) = -\mathbf{B}\vec{\mu}(t), \tag{F3}$$

which establishes the first component of Eq. (F2). Note that $\mathbf{A}\mathbf{A}^+\mathbf{P} = \mathbf{P}$, since $\mathbf{A}^+$ does not act on eigenvectors of $\mathbf{A}$ of eigenvalue zero. Using Newton's equations (Eq. (9)), we have

$$i\dot{\vec{\mu}}(t) = -i\mathbf{B}^+\mathbf{P}\ddot{\vec{y}}(t) = i\mathbf{B}^+\mathbf{P}\mathbf{A}\vec{y}(t) = i\mathbf{B}^+\mathbf{A}\mathbf{P}\vec{y}(t) = i\mathbf{B}^\dagger\mathbf{A}^+\mathbf{A}\mathbf{P}\vec{y}(t) = i\mathbf{B}^\dagger\mathbf{P}\vec{y}(t) = i\mathbf{B}^\dagger\vec{\nu}(t), \tag{F4}$$

which establishes the second component in Eq. (F2).

Hence, evolution under $\mathbf{H}$ also allows us to solve the following variation of Problem 1 and 4:

Problem 8. *Let $\mathbf{A} \succcurlyeq 0$ be an $N \times N$ real-symmetric, PSD, and d-sparse matrix, and $\mathbf{B}$ an $N \times M$ matrix that satisfies $\mathbf{B}\mathbf{B}^\dagger = \mathbf{A}$. Define the normalized state*

$$|\psi(t)\rangle := \frac{1}{\sqrt{2F}} \begin{pmatrix} \mathbf{P}\vec{y}(t) \\ -i\mathbf{B}^+\mathbf{P}\dot{\vec{y}}(t) \end{pmatrix}, \tag{F5}$$

where $F > 0$ is a constant, $\mathbf{P}$ is the projector onto the subspace orthogonal to the null space of $\mathbf{A}$, and $\mathbf{B}^+$ is the Moore-Penrose pseudo-inverse of $\mathbf{B}$. Assume we are given oracle access to $\mathbf{A}$ and to a unitary $\mathcal{W}$ that prepares the initial state $|\psi(0)\rangle$. Given t and ϵ, the goal is to output a state that is ϵ-close to $|\psi(t)\rangle$ in Euclidean norm.

Note that the normalizing constant is

$$F = \frac{1}{2}\vec{y}(t)^T\mathbf{P}\vec{y}(t) + \frac{1}{2}\dot{\vec{y}}(t)^T\mathbf{P}(\mathbf{B}^+)^\dagger\mathbf{B}^+\mathbf{P}\dot{\vec{y}}(t), \tag{F6}$$

which satisfies

$$\begin{aligned}
2\dot{F} &= \dot{\vec{y}}(t)^T\mathbf{P}\vec{y}(t) + \vec{y}(t)^T\mathbf{P}\dot{\vec{y}}(t) + \ddot{\vec{y}}(t)^T\mathbf{P}(\mathbf{B}^+)^\dagger\mathbf{B}^+\mathbf{P}\dot{\vec{y}}(t) + \dot{\vec{y}}(t)^T\mathbf{P}(\mathbf{B}^+)^\dagger\mathbf{B}^+\mathbf{P}\ddot{\vec{y}}(t) \\
&= \dot{\vec{y}}(t)^T\mathbf{P}\vec{y}(t) + \vec{y}(t)^T\mathbf{P}\dot{\vec{y}}(t) - \vec{y}(t)^T\mathbf{A}\mathbf{P}(\mathbf{B}^+)^\dagger\mathbf{B}^+\mathbf{P}\dot{\vec{y}}(t) - \dot{\vec{y}}(t)^T\mathbf{P}(\mathbf{B}^+)^\dagger\mathbf{B}^+\mathbf{P}\mathbf{A}\vec{y}(t) \\
&= \ddot{\vec{y}}(t)^T\mathbf{P}\vec{y}(t) + \vec{y}(t)^T\mathbf{P}\dot{\vec{y}}(t) - \vec{y}(t)^T\mathbf{P}\mathbf{A}(\mathbf{B}^+)^\dagger\mathbf{B}^+\mathbf{P}\dot{\vec{y}}(t) - \dot{\vec{y}}(t)^T\mathbf{P}(\mathbf{B}^+)^\dagger\mathbf{B}^+\mathbf{A}\mathbf{P}\vec{y}(t) \\
&= \dot{\vec{y}}(t)^T\mathbf{P}\vec{y}(t) + \vec{y}(t)^T\mathbf{P}\dot{\vec{y}}(t) - \vec{y}(t)^T\mathbf{P}\mathbf{A}(\mathbf{A}^+)^\dagger\mathbf{B}\mathbf{B}^+\mathbf{P}\dot{\vec{y}}(t) - \dot{\vec{y}}(t)^T\mathbf{P}(\mathbf{B}^+)^\dagger\mathbf{B}^\dagger\mathbf{A}^+\mathbf{A}\mathbf{P}\vec{y}(t) \\
&= \dot{\vec{y}}(t)^T\mathbf{P}\vec{y}(t) + \vec{y}(t)^T\mathbf{P}\dot{\vec{y}}(t) - \vec{y}(t)^T\mathbf{P}\mathbf{B}\mathbf{B}^+\mathbf{P}\dot{\vec{y}}(t) - \dot{\vec{y}}(t)^T\mathbf{P}(\mathbf{B}^+)^\dagger\mathbf{B}^\dagger\mathbf{P}\vec{y}(t) \\
&= \dot{\vec{y}}(t)^T\mathbf{P}\vec{y}(t) + \vec{y}(t)^T\mathbf{P}\dot{\vec{y}}(t) - \vec{y}(t)^T\mathbf{P}\dot{\vec{y}}(t) - \dot{\vec{y}}(t)^T\mathbf{P}\vec{y}(t) \\
&= 0,
\end{aligned} \tag{F7}$$

where we used $\mathbf{B}\mathbf{B}^+\mathbf{P} = \mathbf{A}\mathbf{A}^+\mathbf{P} = \mathbf{P}$. Then, F is independent of t and refers to another conserved quantity different from E. We can use our quantum algorithm to prepare $|\psi(t)\rangle$ in Eq. (F5) by simulating $\mathbf{H}$ in Eq. (12). The complexity of Hamiltonian simulation is the same as that in Thm. 1.

In the encoding used for Problem 8, some amplitudes are proportional to $\mathbf{P}\vec{y}(t)$, and we have more direct access to the displacements of the oscillators than that using the encoding in Problem 1. However, preparing $|\psi(0)\rangle$ in this case is expected to be more costly as it involves the action of $\mathbf{B}^+$. This is the reason why the complexity in the quantum algorithm of Ref. [11] for simulating the wave equation can be dominated by that of the initial state preparation. That case is a special instance of Problem 8 where $\mathbf{A}$ corresponds to the (discretized) Laplacian.

Appendix G: Related work and optimality of our approach

Our quantum algorithm provides the solution to a second-order differential equation, i.e., $\ddot{\vec{y}}(t) = -\mathbf{A}\vec{y}(t)$, encoded in the amplitudes of a quantum state $|\psi(t)\rangle$ defined in Problems 1 and 4. We do this via the classical-to-quantum reduction given in Sec. III, which results in a first-order differential equation, corresponding to a Schrödinger equation with a time-independent Hamiltonian $\mathbf{H}$ that depends on $\mathbf{B}$, where $\mathbf{A} = \mathbf{B}\mathbf{B}^\dagger$. Prior works have also considered quantum algorithms for solving first-order differential equations using a variety of approaches. A prominent example is Ref. [5] and related results (cf. [51, 52]) that use the quantum algorithm for linear systems of equations [8, 53, 54]. Here we argue that those approaches, while possibly applicable to our problem, generally yield inefficient quantum algorithms for Problems 1, 2, 4, and 5 due to issues arising from the encoding. Indeed, by formulating the problem as a Hamiltonian simulation problem, and using optimal methods for the latter, we argue that our approach is the optimal one for solving these problems. Among other useful features, our approach does not necessitate a clock register to encode the solution at all times as in Ref. [5]. More details follow.

One standard approach to formulate the second-order differential equation as a first-order one in our case would be to consider

$$\dot{\vec{v}}(t) = \begin{pmatrix} \mathbf{0} & -\mathbf{A} \\ \mathbb{1}_N & \mathbf{0} \end{pmatrix} \vec{v}(t) \ , \tag{G1}$$

where

$$\vec{v}(t) := \begin{pmatrix} \dot{\vec{y}}(t) \\ \vec{y}(t) \end{pmatrix} \tag{G2}$$

is a vector in $\mathbb{R}^{2N}$ that encodes the coordinates of all oscillators. (We assume that the choice of units is set from the beginning so that calculations are done with real numbers.) Another standard approach would be to consider

$$\dot{\vec{w}}(t) = \begin{pmatrix} \mathbf{0} & -\mathbb{1}_N \\ \mathbf{A} & \mathbf{0} \end{pmatrix} \vec{w}(t) \ , \tag{G3}$$

where

$$\vec{w}(t) := \begin{pmatrix} \dot{\vec{y}}(t) \\ \mathbf{A}\vec{y}(t) \end{pmatrix} \tag{G4}$$

is also a vector in $\mathbb{R}^{2N}$. Equations (G1) and (G3) are also homogeneous first-order differential equations, whose solutions can be obtained by applying an exponential operator to $\vec{v}(0)$ or $\vec{w}(0)$. Note that $\frac{d}{dt}\|\vec{v}(t)\| \neq 0$ and $\frac{d}{dt}\|\vec{w}(t)\| \neq 0$ in general, so we cannot directly apply Hamiltonian simulation methods using these encodings, as the evolution of these vectors is not unitary.

To solve differential equations that do not conserve the norm, Refs. [5, 51, 52] propose a range of quantum algorithms that, for this application, would output (normalized) quantum states

$$|\vec{v}(t)\rangle := \frac{1}{\sqrt{\|\dot{\vec{y}}(t)\|^2 + \|\vec{y}(t)\|^2}} \begin{pmatrix} \dot{\vec{y}}(t) \\ \vec{y}(t) \end{pmatrix} \ , \tag{G5}$$

or

$$|\vec{w}(t)\rangle := \frac{1}{\sqrt{\|\dot{\vec{y}}(t)\|^2 + \|\mathbf{A}\vec{y}(t)\|^2}} \begin{pmatrix} \dot{\vec{y}}(t) \\ \mathbf{A}\vec{y}(t) \end{pmatrix} . \tag{G6}$$

In Refs. [5, 51, 52] this is done by approximating the solutions to the differential equations as solutions to systems of linear equations. The complexity of this approach depends on several parameters, in particular the condition number of the matrix to be inverted. In Ref. [13] a solution is found by giving the solution to the differential equation in exponential form, and then using the LCU approach to approximate the exponential; a related approach that approximates the exponential operator is given in Ref. [55]. That approach is not applicable here, because it requires the matrix to be normal. Here the matrices – the $2{\times}2$ block matrices including $\mathbf{A}$ – are not normal, so the approach cannot be used. Nevertheless, even disregarding such difficulties, we show that this way of encoding the coordinates of the oscillators as in Eqs. (G1) or (G3) is readily problematic.

To observe this, it suffices to consider the evolution of a single normal mode of frequency ω_k, i.e., the eigenvector of $\mathbf{A}$ of eigenvalue $(\omega_k)^2$. We obtain

$$\vec{y}(t) = \vec{a}_k \cos(t\omega_k + \phi_k) , \tag{G7}$$

where $\phi_k \in \mathbb{R}$ is the initial phase and $\vec{a}_k \in \mathbb{R}^N$ is the eigenvector of $\mathbf{A}$. Computing the time average we obtain

$$\langle \|\vec{y}(t)\|^2 \rangle = \frac{1}{2} \|\vec{a}_k\|^2 \tag{G8}$$

and

$$\begin{aligned} \langle \|\dot{\vec{y}}(t)\|^2 \rangle &= \frac{1}{2} |\omega_k|^2 \|\vec{a}_k\|^2 \\ &= |\omega_k|^2 \langle \|\vec{y}(t)\|^2 \rangle . \end{aligned} \tag{G9}$$

(The latter is also the time average of the kinetic energy.) In a case where $\mathbf{A}$ gives rise to normal modes of low frequency, we have $|\omega_k| \ll 1$ (in the corresponding units), and then

$$\langle \|\dot{\vec{y}}(t)\|^2 \rangle \ll \langle \|\vec{y}(t)\|^2 \rangle . \tag{G10}$$

The implication is that the support of $|\vec{v}(t)\rangle$ is, on average, mostly concentrated on the subspace spanned by basis vectors that encode $\vec{y}(t)$. These do not encode the terms appearing in the kinetic energy or $\dot{\vec{y}}(t)$. Hence, the complexity of solving Problem 2 using this encoding, where $|\vec{v}(t)\rangle$ is prepared rather than $|\psi(t)\rangle$, is at least linear in $1/|\omega_k|$, which is the factor needed to increase the desired amplitudes to a constant. This can become unbounded if $|\omega_k| \to 0$.

In the second case we note that the support of $|\vec{w}(t)\rangle$ is, on average, mostly concentrated on the subspace spanned by basis vectors that encode $\dot{\vec{y}}(t)$ instead. That is, the case $\langle \|\mathbf{A}\vec{y}(t)\|^2 \rangle \ll \langle \|\dot{\vec{y}}(t)\|^2 \rangle$ can occur when there are normal modes of low frequency. To solve Problem 1 or Problem 4, and especially Problem 5, the second component of the state must be proportional to $\mathbf{B}^\dagger \vec{y}(t)$ or $\sqrt{\mathbf{A}}\vec{y}(t)$, which requires the application of the pseudo-inverse of $\mathbf{B}$ or $\sqrt{\mathbf{A}}$ to $|\vec{w}(t)\rangle$. The condition number of these pseudo-inverses is also polynomial in $1/|\omega_k|$; for example $\|(\vec{a}_k)^T \mathbf{B}\| = |\omega_k| \ll 1$ for low frequencies. This implies that the complexity of solving Problem 1, Problem 4, or Problem 5 using this encoding, where $|\vec{w}(t)\rangle$ is prepared rather than $|\psi(t)\rangle$, is at least linear in $1/|\omega_k|$, which is the complexity of the most efficient methods to implement the pseudo-inverse [53]. As in the previous case, this can also become unbounded if $|\omega_k| \to 0$.

Similar observations follow directly from the results for the complexity in Refs. [5, 51]. There the complexity is polynomial in the condition number of the matrix that diagonalises the matrix appearing in the differential equations, which is here that with the blocks $-\mathbf{A}$ and $\mathbb{1}_N$ in Eq. (G1), or the negative of these for $\vec{w}(t)$ in Eq. (G3). The matrix that diagonalises this matrix has condition number

$$\max(|\omega_k|, |\omega_k|^{-1}) . \tag{G11}$$

Thus it will become unbounded when $|\omega_k| \to 0$, as described above. Technically, we cannot use the result as given in Ref. [5] directly, because the stability condition it uses requires the matrices appearing in the differential equations to not have eigenvalues exactly on the imaginary axis, as is the case here.

An alternative approach to avoid dependence on the condition number of the diagonalising matrix is that using the log-norm [52]. There we would need to subtract the identity times the log-norm from the original matrix. Here, the log-norm is

$$\max |1 - \omega_k^2|/2 \,. \tag{G12}$$

This means it is always positive, and so subtracting a multiple of the identity would result in a norm that exponentially decreases. Hence the approach of Ref. [52] would yield a complexity that is exponential in time.

At a high level, our encoding is motivated by the conservation of energy in time, and places equal emphasis on those terms that encode the kinetic energy $(\dot{\vec{y}}(t))$ and the potential energy $(\mathbf{B}^\dagger \vec{y}(t))$ of the oscillators. The other two encodings do not have this feature: $|\vec{v}(t)\rangle$ underrepresents the kinetic energy terms and $|\vec{w}(t)\rangle$ underrepresents the potential energy terms, bringing the issues discussed above.

Last, we provide a comment on the relation between our results and a closely related result in Ref. [11] for simulating the wave equation. In Ref. [11] it is shown that the wave equation, a second-order and homogeneous differential equation, can also be mapped to a Schrödinger equation. Their construction is related to ours in that it uses a factorization of the (discrete) Laplacian as $L = \mathbf{BB}^\dagger$, and indeed it is well-known that the wave equation is one example of a classical system of coupled oscillators where all masses and springs are uniform, and where the couplings between oscillators are geometrically local (e.g., on the square grid). However, the encoding used in Ref. [11] is different to ours; for example, N amplitudes are reserved to encode the intensity of the wave, which corresponds to $\vec{y}(t)$ in our case. (The component of $\vec{y}(t)$ on the eigenvector of eigenvalue 0 of L is projected out.) The other amplitudes are proportional to $\mathbf{B}^+ \dot{\vec{y}}(t)$, where $\mathbf{B}^+$ is the pseudo-inverse of $\mathbf{B}$. That is, the state prepared by the algorithm of Ref. [11] is of the form

$$|\phi(t)\rangle \propto \begin{pmatrix} \vec{y}(t) \\ \mathbf{B}^+ \dot{\vec{y}}(t) \end{pmatrix} \,, \tag{G13}$$

where the constant of proportionality is also time-independent. This is essentially the same encoding as the one discussed in Appendix F, since $\mathbf{P}\vec{y}(t) = \vec{y}(t)$ by assumption. The length of this vector is preserved in time since the evolution is unitary. Nevertheless, one implication of using this encoding to solve Problem 1 or Problem 2 for this example is the need to invert $\mathbf{B}^+$ initially, to obtain amplitudes proportional to $\mathbf{B}^+ \vec{y}(0)$. The condition number can be large, i.e., it is $\mathcal{O}(\text{poly}(N))$ for the example of the wave equation, and hence the initial state preparation step $(t = 0)$ can be inefficient. Indeed, Ref. [11] manages to give evidence of a polynomial quantum speedup only. In addition, when considering the wave equation, the system is geometrically local and implies that N is polynomial in the evolution time t (i.e, t is exponential in n). In our problem, however, we can treat a significantly larger set of instances: we do not impose uniform masses, uniform spring constants, or even geometrically-local interactions, and we can allow for times $t = \mathcal{O}(\text{poly}(n))$. This generality together with our improved encoding are key to obtaining our claimed exponential quantum speedups for the problems defined.