Preparing Arbitrary Continuous Functions in Quantum Registers With Logarithmic Complexity

Arthur G. Rattew\textsuperscript{1,2,†} and Bálint Koczor\textsuperscript{1,∗}

\textsuperscript{1}Department of Materials, University of Oxford, Parks Road, Oxford OX1 3PH, United Kingdom
\textsuperscript{2}JPMorgan Chase Bank N.A., Future Lab for Applied Research and Engineering

Quantum computers will be able solve important problems with significant polynomial and exponential speedups over their classical counterparts, for instance in option pricing in finance, and in real-space molecular chemistry simulations. However, key applications can only achieve their potential speedup if their inputs are prepared efficiently. We effectively solve the important problem of efficiently preparing quantum states following arbitrary continuous (as well as more general) functions with complexity logarithmic in the desired resolution, and with rigorous error bounds. This is enabled by the development of a fundamental subroutine based off of the simulation of rank-1 projectors. Combined with diverse techniques from quantum information processing, this subroutine enables us to present a broad set of tools for solving practical tasks, such as state preparation, numerical integration of Lipschitz continuous functions, and superior sampling from probability density functions. As a result, our work has significant implications in a wide range of applications, for instance in financial forecasting, and in quantum simulation.

I. INTRODUCTION

With rapidly accelerating progress in hardware development, quantum computers are quickly becoming a realistic technology. New records are constantly being set, demonstrating that these machines are capable of significantly outperforming the best classical computers in certain settings \cite{1,2,3}. As the technology is scaled up, it will ultimately enable powerful quantum algorithms capable of both polynomial and exponential speedups over their classical counterparts. However, many of these algorithms require efficient state preparation procedures to achieve their speedup, as is the case in: option pricing \cite{4,5}, machine learning \cite{6,7,8,9}, matrix inversion \cite{10}, quantum chemistry \cite{11,12,13}, and quantum Monte Carlo based algorithms \cite{14,15,16}. As such, it remains a crucial open question: can we efficiently prepare quantum states given a set of prespecified amplitudes?

Unfortunately, a quantum circuit capable of producing an arbitrary quantum state necessitates exponential complexity in general \cite{17}, although recent works have shown that the exponential circuit depth cost can actually be made linear in exchange for utilizing an exponential number of ancillary qubits \cite{18,19}. As a result, in practice, it is essential to exploit specific properties of the state being produced.

In this work, we effectively solve the important problem of efficiently preparing continuous (as well as some more general) functions in quantum registers, and we do so with rigorous theoretical guarantees. While we illustrate the core subroutine underpinning our novel technique in Fig. 1, our main result is stated as Theorem 1. Succinctly and informally stated, we prove that our approach has a constant query complexity (asymptotically in the number of qubits), and obtains a bounded error which can be made arbitrarily small. A remarkable feature of our approach is that the performance of our algorithm becomes asymptotically independent of the qubit count as we increase the number of qubits, and that we can reach this asymptotic independence exponentially quickly in the number of qubits.

The constant cost of our algorithm depends inverse-polynomially on the filling ratio of the function – a quantity that we define in a subsequent section and illustrate in Fig. 2 as the ratio of blue area to the area of the total bounding box. While it is possible to construct pathological functions for which the filling ratio is arbitrarily small, for realistic functions encountered in practice this quantity is typically very reasonable (e.g. usually around $10^{-1}$ to $10^{-3}$). We additionally suggest ways that this constant cost can be made tractable even in pathological cases. Moreover, if future works improve the error-dependence of our simulation subroutine, the impact of the filling ratio will be even further negated.

The implications of our work go significantly beyond just enabling the speedup of existing algorithms via efficient state preparation. Functions have been used for centuries to model physical, chemical, biological, and financial phenomena, and have played crucial roles in the industrial revolution and in modern engineering. As the use of analytical techniques for functions modelling realistic systems rapidly becomes intractable, numerical techniques often become the only available option \cite{20}. As a result, computational tools accelerating such numerical analyses are of broad and vital importance. To this end, the techniques we develop in this paper not only allow for the preparation of broad classes of functions in quantum registers, but also enable a variety of tools such as for numerical integration of Lipschitz continuous functions, for the optimization of continuous functions, and even for improved sampling from probability density functions.

We begin by reviewing related work. In the next sec-
functions. In particular, given an arbitrary continuous function \( f : [a, b] \rightarrow \mathbb{C} \) on the closed interval \([a, b]\), we wish to prepare the \( n\)-qubit normalized quantum state
\begin{equation}
|\psi\rangle = \frac{1}{\mathcal{N}} \sum_{j=0}^{N-1} f(x_j)|j\rangle,
\end{equation}
where \( \mathcal{N} \) is the normalization factor, \( N = 2^n \), \( |j\rangle \) is a standard basis vector with \( j \in \{0, 1, \ldots, N-1\} \), and \( x_j \) is the \( j^{th} \) grid point in a uniform discretization of the interval \([a, b]\)—while we later remark that non-uniform grid spacings may be advantageous. In general, preparing an arbitrary state in a Hilbert space of \( n \) qubits requires exponential complexity as the dimension of the space grows exponentially in \( n \) \([17, 20]\). However, the algorithm we are about to present demonstrates that the very general set of states of the form shown in Eq. (1) (i.e., those corresponding to discretized continuous functions) can be prepared efficiently.

In the Appendix, we present our results for the analogous, but special, case of efficiently integrable probability density functions as in the case of the Grover-Rudolph \([27]\) approach. However, for brevity, here we focus only on the integration-free case where the state is prepared via samples of \( f \) rather than through its integration. Moreover, we note that the point-wise state preparation approach is strictly more general than integration-based approaches.

Below we present a rank-1 simulation procedure which is both a fundamental subroutine and a primary contribution of this paper. In summary, we establish how the time evolution \( e^{-it\mathcal{H}} \) under the dense rank-one matrix \( \mathcal{H} \propto |\psi\rangle\langle \psi| \) can be efficiently implemented. Simulating this time evolution is a powerful resource and can be combined with diverse tools from quantum information processing to prepare the quantum states \(|\psi\rangle\) effi-
efficiently. For instance, we consider phase estimation and the Hadamard test in the appendix, but this subroutine also enables other very useful algorithms, e.g. for efficiently integrating Lipschitz continuous functions. Nevertheless, in the main text we focus on the conceptually most straightforward variant for preparing states of the form in Eq. (1), based on adiabatic evolution.

We first formally define the filling ratio of the function being prepared as it is intrinsic to the asymptotic complexity of our state preparation procedures. We then informally state our main result for the adiabatic procedure, and then describe the algorithm.

As stated in Definition 3, the filling ratio \( F \) of function \( f \) is defined as \( F := \|f\|_1/\|f\|_{\text{max}} \) and has a value \( 0 \leq F \leq 1 \). Intuitively, it can be understood as the integral of the absolute function (i.e. \(|f(x)|\)) divided by the area of the box bounding the absolute function (i.e. the box with width \( b-a \) and height \( \max_{x \in [a,b]} |f(x)| \)). The filling ratio is clearly pictured for some common functions in Figure 2, as the ratio under the curves, to the total area of the bounding box. Finally, we note that we formally define the notion of efficient computability in Section II A as well as in Definition 4.

**Theorem 1** (informal version of Theorem 2). For an efficiently computable function \( f \), our approach prepares the \( n \)-qubit state \( |\psi\rangle = \frac{1}{N} \sum_{j=0}^{N-1} f(x_j) |j\rangle \) with \( N = 2^n \) up to error \( \epsilon \) with query complexity \( O\left(Fp/\epsilon^2\right) \), where \( F \) is the filling ratio and \( p = -4 \). The error \( \epsilon \) is the deviation from an ideal unitary in terms of a spectral distance. The algorithm uses \( O(n + d) \) ancillary qubits (where \( d \) is the number of digits used in the discretization of \( f \)) and has a probability of failure bounded by \( O(\epsilon^2) \).

It is worth additionally noting that the success probability only enters our bound to simplify our proofs — in practice our approach effectively succeeds with probability 1. This is discussed comprehensively in the Appendix.

To enable the adiabatic state preparation procedure, we define a parameterized family of continuous functions \( f_s := (1-s)f_0 + sf_1 \), where \( f_0 \propto 1 \) (i.e. corresponds to an easy-to-prepare state) and \( f_1 \) corresponds to the state we wish to prepare (i.e. \( f_1 := f \)) with \( 0 \leq s \leq 1 \). We then define the parameterized quantum state, \( |\psi_s\rangle \propto \sum_{j=0}^{N-1} f_s(x_j)|j\rangle \). The procedure begins by creating the state \( |\psi_0\rangle \), which by construction of \( f_0 \) is simply \(+\rangle^{\otimes n}\). We then slowly adiabatically morph this starting state to the final state \( |\psi_1\rangle \) by evolving according to the rank-1 projector \( \mathcal{H}(s) \propto |\psi_s\rangle\langle\psi_s| \). Here, one must pick the total evolution time \( T \) so as to satisfy the adiabatic theorem and ensure that the state remains in the ground state of the morphing Hamiltonian \( \mathcal{H}(s) \). As we prove in the Appendix \( T \) is constant bounded (and can therefore be selected utilizing that bound). Of course, we can’t implement a continuous time evolution in practice, and so we discretize the temporal grid uniformly into \( r \) time-steps. In the \( j \)-th time-step, we evolve according to the time-independent Hamiltonian \( \mathcal{H}(\frac{j}{r}) \), yielding the overall evolution:

\[
e^{-iT\mathcal{H}(\frac{j}{r})}e^{-iT\mathcal{H}(\frac{j-1}{r})} \ldots e^{-iT\mathcal{H}(\frac{0}{r})},
\]

As shown in Appendix C this discretization has constant bounded error, which can be made arbitrarily small. Naturally, it is important to explain how each evolution \( e^{-iT\mathcal{H}(\frac{j}{r})} \) is implemented, and we do so in Section II A.

However, we first explain the conceptual difference between “conventional” adiabatic quantum computation, and the approach we have just described. In the standard formulation of adiabatic quantum computation, one defines the interpolated Hamiltonian \( \mathcal{H}(s) = (1-s)\mathcal{H}_0 + s\mathcal{H}_1 \), where \( \mathcal{H}_0 \) has an easy to prepare ground-state, and \( \mathcal{H}_1 \) encodes the problem of interest \( \mathcal{P} \). However, in this definition the energy gap often vanishes (or becomes exponentially small), resulting in the total evolution time \( T \) becoming intractable. By defining our interpolated Hamiltonian as the rank-1 projector \( \mathcal{H}(s) \propto |\psi_s\rangle\langle\psi_s| \), we circumvent this limitation. Indeed, in the appendix we prove that the spectral gap \( g(s) \) is generally lower bounded by a constant, e.g., \( g(s) \geq 1/2 \). This ensures that our total evolution time required is independent of the problem size \( n \).

### A. Efficient Implementation of the Time Evolution

Before describing our time evolution procedure, we outline our oracle query model. Our procedure allows the preparation of a quantum state following an arbitrary continuous distribution, so long as the function is efficiently computable. Precisely, by efficiently computable, we mean that given an input \( x \) represented with \( n \) bits of precision, there exists a classical (and thus a quantum algorithm) returning \( f(x) \) with \( d \) bits of precision with complexity scaling as \( \text{poly}(n,d) \). Equivalently, an efficiently computable function is any function with the oracle implementation \( O_f \) efficiently performing the mapping \( O_f|x\rangle = |x\rangle|f(x)\rangle \). See Definition 4. Without a loss of generality, both the input and output of \( f \) are given in a binary encoding. Intuitively, if there is an efficient classical program (e.g. in Python) evaluating \( f(x) \), then in principle, the function is efficiently computable. Indeed optimised, efficient quantum-arithmetic procedures are already available \[34\]. Classically, to evaluate the function \( f \) with \( N \) distinct inputs would require \( O(N) \) queries to the function – in the quantum setting, only \( O(1) \) queries are required. This exponential advantage is intuitively suggestive of the potential power of the applications utilizing this procedure.

Furthermore, the time complexity of our approach is ultimately determined by the circuit depth. Given in most practical cases the function values can be computed via arithmetic operations, we expect that the oracle implementation requires less than a quadratic circuit depth (in terms of \( n \) and the number of digits \( d \) used in the discretization of \( f \) \[34\].
Given access to this oracle, we utilise the approach of Rebentrost et al. [22] to efficiently simulate quantum dynamics under the rank-1 matrix $A(s)$ whose entries are given by $[A(s)]_{kl} := f_k(x_k)f^*_l(x_l)$. Note that the function values $f_k := (1 - s)f_0 + sf_1$ can be computed by an oracle, but our definition also permits storing $f_0(x_j)$ and $f_1(x_j)$ in QRAM and computing linear combinations thereof using only addition and multiplication. Depending on the QRAM model, this could potentially make the cost of the oracle query $O(1)$, and thus the algorithm’s circuit depth would be the same as its query complexity.

We proceed by encoding the matrix $A(s) \in \mathbb{C}^{N \times N}$ into a quadratically larger one-sparse matrix $S_A \in \mathbb{C}^{N^2 \times N^2}$—wherby a one-sparse matrix contains only a single non-zero entry in each row/column. This property allows us to utilise powerful one-sparse simulation techniques [24-26] thereby exactly simulating the time evolution $e^{-iSA}$ which acts on the $n$-qubit main register as well as an $n$-qubit ancillary register $-$ and the simulation requires a further $d$-qubit, and two 1-qubit ancilla registers. After measuring and discarding the ancilla register we implement the mapping

$$U''(\Delta t, s)\ket{\psi_0} = e^{-i\Delta t H(s)}\ket{\psi_0} + \ket{\mathcal{E}}$$

under the effective rank-1 Hamiltonian $H(s) := A(s)/N$ up to a bounded error term $\|\mathcal{E}\| \in O(\Delta t^2)$. By setting the time step sufficiently small $\Delta t \ll 1$ we can simulate the evolution $U''(\Delta t, s) = e^{-i\Delta t H(s)}$ up to an arbitrarily small error. This allows us to implement the piecewise constant adiabatic evolution for overall time $T$ in $r$ iterations (timesteps) as per Eq. (2). This has the added benefit of also decreasing the adiabatic discretization error as we increase $r$. In the Appendix, we prove that applying this unitary evolution to the initial state $\ket{+}^\otimes n$ indeed maps onto our desired state $\ket{\psi}$ with a bounded error that only depends on properties of the encoded function.

### Table I: Filling ratios (as illustrated in Fig. 2) for common probability distributions as a function of their parameters.

| Normal distribution | Log-normal distribution |
|---------------------|------------------------|
| $\mu$               | $\sigma$               | filling ratio $\mathcal{F}$ | filling ratio $\mathcal{F}$ |
| 1/2                 | 0.1                    | $2.5 \times 10^{-1}$        | 0.5                     | $5.5 \times 10^{-1}$ |
| 1/2                 | 0.05                   | $1.3 \times 10^{-1}$        | 1.0                     | $7.6 \times 10^{-1}$ |
| 1/2                 | 0.01                   | $2.5 \times 10^{-2}$        | 1.5                     | $6.1 \times 10^{-3}$ |

| Slater function     | Riemann zeta function  |
|---------------------|------------------------|
| $\alpha$            | filling ratio $\mathcal{F}$ |
| 5                   | 3.7 $\times 10^{-1}$    |
| 10                  | 2.0 $\times 10^{-1}$    |
| 20                  | 1.0 $\times 10^{-1}$    |
| 100                 | 3.1 $\times 10^{-1}$    |

B. Limitations

Our approach incurs two types of algorithmic error. First, we approximate an idealised continuous adiabatic evolution through a finite series of $r$ time-independent evolutions. Second, our Hamiltonian simulation also incurs algorithmic error, as previously mentioned. However, both kind of errors can be suppressed arbitrarily by increasing the number of iterations $r$ and thus decreasing the time each time-independent unitary is evolved for (i.e. by increasing the resolution in the temporal discretization).

While our approach is very general and is also applicable to non-continuous general functions (mappings), its main practical limitation is that its complexity depends on the filling ratio $\mathcal{F}$ of the function. In practice, we expect this filling ratio to be a modest constant. See Table I for filling ratios of common probability distributions and other functions. Nevertheless, we can easily construct artificial, pathological instances where $\mathcal{F}$ can be arbitrarily small. As a result, while not essential, improving the algorithm’s asymptotic dependence on $\mathcal{F}$ is a promising direction for future research.

For an example of a pathological instance that helps build intuition as to the limitation of $\mathcal{F}$, consider the unstructured search problem where we are given a function $F^{(n)} : \{0, 1\}^n \rightarrow \{0, 1\}$ (where only one input $x = m$ corresponds to a non-zero output). Preparing the function $F^{(n)}$ in an $n$-qubit quantum register would reveal the non-zero value at $m$ upon measuring the state and thus would solve the unstructured search problem with high probability [38, 39]. While this is not a discretized (continuous) function, it is directly analogous to a box function of width $\approx 2^{-n}$ depending on the qubit count $n$. This results in a generalised filling ratio $\mathcal{F} \approx 2^{-n/2}$ that decreases exponentially as we detail in Appendix C.
and thus our approach would require $O(2^n)$ calls to the oracle $f(x)$ and is thus less efficient than a classical direct search. Note that such a pathological problem instance was only constructed because the generalised filling ratio $\mathcal{F}$ is not an asymptotic constant (while the special-case filling ratio $\mathcal{F}$ for continuous functions is an asymptotic limit in $n$ with guaranteed convergence to a constant).

Finally, the impact of the filling ratio may in practice be circumvented. In one scenario, suppose the function being prepared has the majority of its density in an interval $[a', b'] \subset [a, b]$ that is relatively small compared to the domain. Then, we can simply prepare the function in $[a', b']$ (at some smaller qubit count) with a proportionally larger filling ratio. We can then add further qubits in the $|0\rangle$ state and shift the resulting function via quantum addition. If the function being prepared is non-zero on an exponentially small portion of the overall state space, and the location of the non-zero states are known, then this technique can yield an exponential improvement in the filling ratio. As a result, functions that are very narrow (i.e. very peaked), but have all of their density within a single small known region, including periodic narrow peaks (e.g. functions approaching a Dirac comb), can have their effective filling ratios increased significantly. We also note that $d$-sparse simulation techniques may also be useful for such very peaked functions if we can specify an oracle that efficiently computes the locations of non-zero function values [40].

Another technique one can employ to mitigate the constant cost of the filling ratio is to utilize a non-uniform grid. In such a grid, more samples of the function are taken in regions where the function is particularly narrow, thus giving a larger effective filling ratio. We leave such non-uniform grids as a topic for future exploration.

III. NUMERICAL DEMONSTRATION

We now demonstrate our approach in the preparation of two single-variate continuous functions which are practically relevant. First, we consider the probability density function of a log-normal distribution in Fig. 3 (a, above) which is relevant in modelling a wide range of natural phenomena as well as financial systems [6, 7, 37]. Second, we consider a Slater-type wave-function as a linear combination of primitive functions $\beta e^{-\alpha (x-x_0)^2}$ with constants $\alpha, \beta, x_0 \in \mathbb{R}$. Gaussian and Fourier approximations to atomic orbitals are primarily used in quantum chemistry applications but despite their paramount practical utility they cannot accurately capture the crucial cusps at positions of atomic nuclei [35]. Indeed our approach allows the efficient preparation of exact Slater-type orbitals thereby increasing the accuracy of grid-based simulations [13, 35].

In Fig. 3 (a) we initially prepare a uniform superposition $|+\rangle^\otimes n$ in a register of $n$ qubits and then adiabatically morph this into our desired function. In the figure,
the color gradient represents the smooth interpolation of $f_s$ throughout the adiabatic trajectory, with the lightest shade representing $f_0$ and the black line representing $f_1$. Taking a slice in the figure where only a certain color is kept (corresponding to a time $s$), would yield a plot of the function $f_s$.

In Fig. 3(b) we explicitly simulate the dynamics under $e^{-iT/R}$ that acts on overall $2n$ qubits thereby taking into account all sources of error (including error in binary encoding of the function values $f(x)$). Here, each line corresponds to a series of simulations evaluated with a fixed time $T$ and a fixed number of iterations $r$ as we increase the number of qubits $n$. Our simulations nicely confirm our bounds in the following two distinct asymptotic limits.

First, as we increase the number of qubits $n$ the state-preparation error (solid lines) rapidly, in exponential order, approaches a constant value as expected from our asymptotic bounds – the asymptotic constant error (dashed lines are extrapolations) is then determined by the filling ratio $F$ and by $T$ and $r$. Second, the series of lines from black to magenta in Fig. 3(b) represent simulations with an increasing $r$ and confirm that as we increase the number of iterations $r$ the error decreases polynomially (as both the precision of the simulation step increases, and the discretization error of the adiabatic evolution decreases).

It is important to note that our error bounds in Theorem I are very pessimistic as we have used an unrealistic, worst-case scenario argument in our proofs. In particular, while our bounds scale as $O(\sqrt{1/r})$, in practically relevant cases the error is expected to be dominated by the algorithmic error of the low-rank simulation approach – as it depends polynomially on the filling ratio. A simple argument shows that neglecting the adiabatic evolution error allows us to obtain the tighter error bound $O(1/r)$. This is nicely confirmed in Fig. 3(c) whereby we plot the asymptotic constant errors from Fig. 3(b) as a function of the number of iterations $r$ and obtain an empirical scaling from approximately $O(r^{-1.19})$ to $O(r^{-1.3})$. We also note that the absolute error of our procedure is below our error bound by many orders of magnitude, suggesting reasonably low absolute factors in practice.

IV. FURTHER APPLICATIONS

The techniques we have developed in this paper allow for a number of promising further applications. Some of these applications are related to the state preparation procedure, and some are consequences of the techniques we have developed. Of course, the adiabatic state preparation procedure that we present immediately enables applications that rely on continuous input states, such as those we have outlined in Section I.

However, the state preparation procedure can actually be an application in its own-right. For instance, an important problem is to draw samples following a particular given continuous probability density function. Classically, techniques to do so, such as inverse transform sampling, generally require integrating the probability density function in question, and thus have complexity that scales with the resolution of the grid discretization $\Omega$. Our state-preparation approach immediately enables such sampling, simply by preparing the square-root of the desired density function and then measuring the resulting state. Our approach has overall logarithmic complexity in doing so (and is thus an instance of exponential quantum advantage), and moreover produces truly random samples from the density function, rather than just pseudo-random samples.

Furthermore, given that our approach allows us to prepare a quantum state whose amplitudes are proportional to the samples $f(x_j)$, it follows that the measurement outcome of the bitstring $j \in \{0,1\}^n$ has a probability proportional to $|f(x_j)|^2$. As a result, we can directly use our approach to efficiently draw importance samples from efficiently computable functions, and if the function has one dominant maximum then importance sampling can even find this global maximum with a high probability.

An alternative proposal for using our techniques for global optimization follows. Given a function $g : \mathbb{R} \mapsto \mathbb{C}$, preparing $f(x) = g(x)^k$ for some constant $k > 1$ allows us to find the global maximum of $g(x)$ with probability that improves exponentially in $k$ through direct sampling. However, the filling ratio of $f(x)$ also decreases exponentially in $k$, and so the complexity of preparing the state negates any benefit of using $k > 1$. On the other hand, if the algorithm’s asymptotic dependence of $F$ can be improved, then such an approach could be an extremely promising global optimization procedure.

Additionally, as discussed in the Appendix, the simulation technique we have developed enables the efficient integration of Lipschitz continuous probability density functions through the use of phase estimation. In particular, we can prepare the state $|\psi\rangle$ with our adiabatic approach and apply time evolution under $\mathcal{H}(s)$. By implementing phase estimation we can estimate the only non-zero eigenvalue of $\mathcal{H}(s)$ with near-perfect success rate and thus obtain an estimate of the $L^2$ norm of the encoded function.

V. DISCUSSION AND CONCLUSION

In this work we introduced a family of techniques that allow the efficient loading of any efficiently computable function onto a quantum register such that amplitudes of the state vector $|\psi\rangle$ are proportional to samples of the function. Moreover, the state preparation technique is actually more general, and also applies to certain discontinuous functions.

Motivating the power of our techniques, the number of states in an $n$-qubit Hilbert space grows exponentially as $2^n$, so a quantum computer allows us to efficiently “compute” an exponential number of samples from our
function with one oracle query. This property enables, amongst other things, efficient state preparation which is essential to a large number of key applications of quantum computers. For example, applications in fields ranging from the simulation of quantum physics and chemistry [13] to financial forecasting [7] will all benefit from our techniques.

A central idea in this paper is that we can embed the desired quantum state $|\psi\rangle$ into a rank-1 Hamiltonian $H \propto |\psi\rangle\langle\psi|$ whose ground state is exactly $|\psi\rangle$, and with all other orthogonal states having an eigenvalue of 0. We then utilized powerful low-rank [22] and 1-sparse [23–26] Hamiltonian simulation techniques from the literature, allowing us to efficiently simulate $H$. Among other variants, we presented an adiabatic-evolution approach that enables quasi-deterministic preparation of the ground state of the rank-1 matrix $H$. Central to the argument of this procedure’s efficiency is that the spectral gap of such a rank-1 matrix is the same as its spectral norm, leading to a rare case where an adiabatic algorithm has a provably bounded spectral gap (which is essential to proving the efficiency of the adiabatic procedure). We rigorously proved that the query complexity of our adiabatic approach is a constant independent of the number of qubits. That is, we need only make a constant number of calls to the oracle computing the function amplitudes, regardless of the number of qubits the state is being prepared on. Moreover, we also prove that the state-preparation error and failure probability can be suppressed arbitrarily by increasing the temporal resolution (i.e. the number of steps in the adiabatic evolution $r$). It is worth noting that even though we perform measurements to reset the state of the ancilla register, our approach becomes reversible for a sufficiently high state-preparation precision given that the measurement outcomes are then non-deterministic as per Theorem 1. Moreover, we stress that the non-deterministic component of the algorithm is essentially an analytical convenience, and that in practice one can accept any outcome of the ancilla measurement.

We would additionally like to emphasize that the presented ideas are ripe for future modification, optimization, and generalization. For example, in practice one can use our adiabatic approach to efficiently and quickly prepare a crude approximation of $|\psi\rangle$ (via low precision $\epsilon$) and then either refine it via our phase estimation or destructive interference procedures (with high probability of success). Moreover, further optimizing the algorithm’s asymptotic dependence on both $\epsilon$ and $F$ is a promising area of future exploration, with potentially significant ramifications. Indeed we expect with minor modifications one may be able to reduce the $O(\Delta t^2)$ error of the low-rank simulation approach – which is currently a bottleneck. As such, we expect our work will spark further developments, and can potentially lead to a rich set of quantum algorithms for manipulating and dealing with continuous functions by utilizing quantum parallelism.

Our work almost immediately allows for the preparation of continuous multivariate functions, albeit with the main limitation that the filling ratio may decrease asymptotically in the dimension of the function being prepared, as we discuss in Appendix C.

The presented algorithms (and related ones in the literature) assume execution on fault-tolerant quantum computers. However, the quantum devices of the near-future are error prone [42] and there is limited work on preparing quantum states for NISQ devices. One possible approach utilizes quantum generative adversarial networks (qGAN) to tune the parameters in variational circuits to load a given distribution [43]. Some work on using qGANs to produce multivariate quantum states has also been conducted [14, 15]. Alternatively, Rattew et al. demonstrate how normal distributions may be efficiently produced in a manner resistant to most hardware errors [46]. The techniques presented in that work can potentially be directly applied to the work presented in this paper to discard state preparation attempts in which hardware errors likely occurred. Furthermore, combination with powerful error mitigation strategies may also be possible, should the states produced be used to compute the expectation values of observables [47]. However, to be useful with the algorithms presented in this work the infidelity of (uncorrected) hardware operations would likely need to be orders of magnitude lower than current state-of-the-art approaches (e.g. refs [48, 49]).

ACKNOWLEDGMENTS

The authors would like to thank Simon Benjamin, Marco Pistoia, Dylan Herman, Yue Sun, and the entire FLARE group at JPMorgan Chase Bank, N.A. for numerous insightful and helpful conversations. The authors would like to give special thanks to Yue Sun and Dylan Herman for carefully checking the proofs. This work was supported in part by JPMorgan Chase Bank, N.A. through the Future Lab for Applied Research and Engineering (FLARE) Ph.D. Fellowship program. B.K. thanks the University of Oxford for a Glasstone Research Fellowship and Lady Margaret Hall, Oxford for a Research Fellowship. The authors would like to acknowledge the use of the University of Oxford Advanced Research Computing (ARC) facility in carrying out this work.

DISCLAIMER

This paper was prepared for information purposes by the teams of researchers from the various institutions identified above, including the Future Lab for Applied Research and Engineering (FLARE) group of JPMorgan Chase Bank, N.A.. This paper is not a product of the Research Department of JPMorgan Chase & Co. or its affiliates. Neither JPMorgan Chase & Co. nor any of its affiliates make any explicit or implied representation or warranty and none of them accept any liability.
in connection with this paper, including, but not limited to, the completeness, accuracy, reliability of information contained herein and the potential legal, compliance, tax or accounting effects thereof. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction.

[1] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, K. Guerin, S. Habegger, M. P. Harrigan, M. J. Hartmann, A. Ho, M. Hoffmann, T. Huang, T. S. Humble, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. V. Klimov, S. Knyshe, A. Korotkov, F. Kostritsa, D. Landhuis, M. Lindmark, E. Lucero, D. Lyakh, S. Mandrà, J. R. McClean, M. McEwen, A. Megramt, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Y. Niu, E. Ostby, A. Petukhov, J. C. Platt, C. Quintana, E. G. Rieffel, P. Roushan, N. C. Rubin, D. Sank, K. J. Satzinger, V. Smelyanskiy, K. J. Sung, M. D. Trevithick, A. Vainsencher, B. Villalonga, T. White, Z. J. Yao, P. Yeh, A. Zalcman, H. Neven, and J. M. Martinis, Quantum supremacy using a programmable superconducting processor, Nature 574, 505 (2019).

[2] H.-S. Zhong, Y.-H. Deng, J. Qin, H. Wang, M.-C. Chen, L.-C. Peng, Y.-H. Luo, D. Wu, S.-Q. Gong, H. Su, Y. Hu, P. Hu, X.-Y. Yang, W.-J. Zhang, H. Li, Y. Li, X. Jiang, L. Gan, G. Yang, L. You, Z. Wang, L. Li, N.-L. Liu, J. J. Renema, C.-Y. Lu, and J.-W. Pan, Phase-Programmable Gaussian Boson Sampling Using Stimulated Squeezed Light, Physical Review Letters 127, 180502 (2021).

[3] Y. Wu, W.-S. Bao, S. Cao, F. Chen, M.-C. Chen, X. Chen, T.-H. Chung, H. Deng, Y. Du, D. Fan, M. Gong, C. Guo, C. Guo, S. Guo, L. Han, L. Hong, H.-L. Huang, Y.-H. Luo, L. Li, N. Li, S. Li, Y. Li, F. Liang, C. Lin, J. Lin, H. Qian, D. Qiao, H. Rong, H. Su, L. Sun, L. Wang, S. Wang, D. Wu, Y. Xu, K. Yan, W. Yang, Y. Yang, Y. Ye, J. Yin, C. Ying, J. Yu, C. Zha, C. Zhang, H. Zhang, K. Zhang, Y. Zhang, H. Zhao, Y. Zhao, L. Zhou, Q. Zhu, C.-Y. Lu, C.-Z. Peng, X. Zhu, and J.-W. Pan, Strong Quantum Computational Advantage Using a Superconducting Quantum Processor, Physical Review Letters 127, 180501 (2021).

[4] S. Ebadi, T. T. Wang, H. Levine, A. Keesling, G. Mazzola, A. Montanaro, Quantum supremacy using a 256-atom programmable quantum simulator, Nature 574, 227 (2021).

[5] M. Gong, S. Wang, C. Zha, M.-C. Chen, H.-L. Huang, Y. Wu, Q. Zhu, Y. Zhao, S. Li, S. Guo, H. Qian, Y. Ye, F. Chen, C. Ying, J. Yu, D. Fan, D. Wu, H. Su, H. Deng, H. Rong, K. Zhang, S. Cao, J. Lin, Y. Xu, L. Sun, C. Guo, N. Li, F. Liang, V. M. Bastidas, K. Nemoto, W. J. Munro, Y.-H. Huo, C.-Y. Lu, C.-Z. Peng, X. Zhu, and J.-W. Pan, Quantum walks on a programmable two-dimensional 62-qubit superconducting processor, Science 372, 948 (2021).

[6] S. Chakrabarti, R. Krishnakumar, G. Mazzola, N. Stamatopoulos, S. Woerner, and W. J. Zeng, A threshold for quantum advantage in derivative pricing, Quantum 5, 463 (2021).

[7] N. Stamatopoulos, D. J. Egger, Y. Sun, C. Zoufal, R. Iten, N. Shen, and S. Woerner, Option pricing using quantum computers, Quantum 4, 291 (2020).

[8] D. Herman, C. Googin, X. Liu, A. Galda, I. Safro, Y. Sun, M. Pistoia, and Y. Alexeev, A survey of quantum computing for finance, arXiv preprint arXiv:2201.02773 (2022).

[9] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum principal component analysis, Nature Physics 10, 631 (2014).

[10] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Physical Review A 98, 032309 (2018).

[11] M. Pistoia, S. F. Ahmad, A. Ajagekar, A. Buts, S. Chakrabarti, D. Herman, S. Hu, A. Jena, P. Minssen, P. Niroula, et al., Quantum machine learning for finance iccad special session paper, in 2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD) (IEEE, 2021) pp. 1–9.

[12] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, Physical review letters 103, 150502 (2009).

[13] H. H. S. Chan, R. Meister, T. Jones, D. P. Tew, and S. C. Benjamin, Grid-based methods for chemistry modelling on a quantum computer, arXiv preprint arXiv:2202.05864 (2022).

[14] N. J. Ward, I. Kassal, and A. Aspuru-Guzik, Preparation of many-body states for quantum simulation, The Journal of chemical physics 130, 194105 (2009).

[15] S. Woerner and D. J. Egger, Quantum risk analysis, npj Quantum Information 5, 1 (2019).

[16] A. Montanaro, Quantum speedup of monte carlo methods, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 471, 20150301 (2015).

[17] M. Plesch and Č. Brukner, Quantum singular-value decomposition of nonsparse low-dimensional 62-qubit superconducting processor, Science 372, 948 (2021).
rank matrices, Physical review A 97, 012327 (2018).
[23] A. M. Childs, Quantum information processing in continuous time, Ph.D. thesis, Massachusetts Institute of Technology (2004).
[24] D. Aharonov and A. Ta-Shma, Adiabatic quantum state generation and statistical zero knowledge, in Proceedings of the thirty-fifth annual ACM symposium on Theory of computing (2003) pp. 20–29.
[25] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. 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.
[26] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Efficient quantum algorithms for simulating sparse hamiltonians, Communications in Mathematical Physics 270, 359 (2007).
[27] L. Grover and T. Rudolph, Creating superpositions that correspond to efficiently integrable probability distributions, arXiv preprint quant-ph/0208112 (2002).
[28] G. Marin-Sanchez, J. Gonzalez-Conde, and M. Sanz, Quantum algorithms for approximate function loading, arXiv preprint arXiv:2111.07933 (2021).
[29] A. Holmes and A. Matsuura, Efficient quantum circuits for accurate state preparation of smooth, differentiable functions, in 2020 IEEE International Conference on Quantum Computing and Engineering (QCE) (IEEE, 2020) pp. 169–179.
[30] A. C. Vazquez and S. Woerner, Efficient state preparation for quantum amplitude estimation, Physical Review Applied 15, 034027 (2021).
[31] S. Herbert, No quantum speedup with grover-rudolph state preparation for quantum monte carlo integration, Physical Review E 103, 063302 (2021).
[32] W. Van Dam, M. Mosca, and U. Vazirani, How powerful is adiabatic quantum computation?, in Proceedings 42nd IEEE Symposium on Foundations of Computer Science (2001) pp. 279–287.
[33] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Quantum computation by adiabatic evolution, arXiv preprint quant-ph/0001106 (2000).
[34] T. H"{a}ner, M. Roetteler, and K. M. Svore, Optimizing quantum circuits for arithmetic, arXiv preprint arXiv:1805.12445 (2018).
[35] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, Quantum computational chemistry, Reviews of Modern Physics 92, 015003 (2020).
[36] P. Borwein, An efficient algorithm for the Riemann zeta function, in Canadian Mathematical Society Conference Proceedings, Vol. 27 (2000) pp. 29–34.
[37] E. L. Crow and K. Shimizu, Lognormal distributions (Marcel Dekker New York, 1987).
[38] L. K. Grover, A fast quantum mechanical algorithm for database search, in Proceedings of the twenty-eighth annual ACM symposium on Theory of computing (1996) pp. 212–219.
[39] M. A. Nielsen and I. Chuang, Quantum computation and quantum information (2002).
[40] D. W. Berry, A. M. Childs, and R. Kothari, Hamiltonian simulation with nearly optimal dependence on all parameters, in 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (IEEE, 2015) pp. 792–809.
[41] S. Olver and A. Townsend, Fast inverse transform sampling in one and two dimensions, arXiv preprint arXiv:1307.1223 (2013).
[42] J. Preskill, Quantum computing in the nisq era and beyond, Quantum 2, 79 (2018).
[43] C. Zoufal, A. Lucchi, and S. Woerner, Quantum generative adversarial networks for learning and loading random distributions, npj Quantum Information 5, 1 (2019).
[44] E. Y. Zhu, S. Johri, D. Bacon, M. Esencan, J. Kim, M. MuiR, N. Murgai, J. Nguyen, N. Pisenti, A. Schouela, et al., Generative quantum learning of joint probability distribution functions, arXiv preprint arXiv:2109.06315 (2021).
[45] G. Agliardi and E. Prati, Optimal tuning of quantum generative adversarial networks for multivariate distribution loading, Quantum Reports 4, 75 (2022).
[46] A. G. Rattew, Y. Sun, P. Minssen, and M. Pistoia, The efficient preparation of normal distributions in quantum registers, Quantum 5, 609 (2021).
[47] B. Koczor, Exponential error suppression for near-term quantum devices, Physical Review X 11, 031057 (2021).
[48] J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, C. Baldwin, M. Foss-Feig, D. Hayes, K. Mayer, C. Ryan-Anderson, et al., Demonstration of the qcd trapped-ion quantum computer architecture, arXiv preprint arXiv:2003.01293 (2020).
[49] E. Pelofske, A. Bärttschi, and S. Eidenbenz, Quantum volume in practice: What users can expect from nisq devices, arXiv preprint arXiv:2203.03816 (2022).
[50] V. Vedral, A. Barenco, and A. Ekert, Quantum networks for elementary arithmetic operations, Physical Review A 54, 147 (1996).
[51] T. G. Draper, Addition on a quantum computer, arXiv preprint quant-ph/0008033 (2000).
[52] T. G. Draper, S. A. Kutin, E. M. Rains, and K. M. Svore, A logarithmic-depth quantum carry-lookahead adder, arXiv preprint quant-ph/0406142 (2004).
[53] Y. Takahashi, S. Tani, and N. Kunihiro, Quantum addition circuits and unbounded fan-out, arXiv preprint arXiv:0910.2530 (2009).
[54] M. K. Bhaskar, S. Hadfield, A. Papageorgiou, and I. Petras, Quantum algorithms and circuits for scientific computing, arXiv preprint arXiv:1511.08253 (2015).
[55] T. Albash and D. A. Lidar, Adiabatic quantum computation, Reviews of Modern Physics 90, 015002 (2018).
[56] F. Casas, A. Murua, and M. Nadnic, Efficient computation of the zassenhaus formula, Computer Physics Communications 183, 2386 (2012).
[57] S. Olver and A. Townsend, Fast inverse transform sampling in one and two dimensions, arXiv preprint arXiv:1307.1223 (2013).
[58] J. Preskill, Quantum computing in the nisq era and beyond, Quantum 2, 79 (2018).
[59] C. Zoufal, A. Lucchi, and S. Woerner, Quantum generative adversarial networks for learning and loading random distributions, npj Quantum Information 5, 1 (2019).
[60] E. Y. Zhu, S. Johri, D. Bacon, M. Esencan, J. Kim, M. MuiR, N. Murgai, J. Nguyen, N. Pisenti, A. Schouela, et al., Generative quantum learning of joint probability distribution functions, arXiv preprint arXiv:2109.06315 (2021).
[61] G. Agliardi and E. Prati, Optimal tuning of quantum generative adversarial networks for multivariate distribution loading, Quantum Reports 4, 75 (2022).
[62] A. G. Rattew, Y. Sun, P. Minssen, and M. Pistoia, The efficient preparation of normal distributions in quantum registers, Quantum 5, 609 (2021).
[63] B. Koczor, Exponential error suppression for near-term quantum devices, Physical Review X 11, 031057 (2021).
[64] J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, C. Baldwin, M. Foss-Feig, D. Hayes, K. Mayer, C. Ryan-Anderson, et al., Demonstration of the qcd trapped-ion quantum computer architecture, arXiv preprint arXiv:2003.01293 (2020).
[65] E. Pelofske, A. Bärttschi, and S. Eidenbenz, Quantum volume in practice: What users can expect from nisq devices, arXiv preprint arXiv:2203.03816 (2022).
[66] V. Vedral, A. Barenco, and A. Ekert, Quantum networks for elementary arithmetic operations, Physical Review A 54, 147 (1996).
[67] T. G. Draper, Addition on a quantum computer, arXiv preprint quant-ph/0008033 (2000).
[68] T. G. Draper, S. A. Kutin, E. M. Rains, and K. M. Svore, A logarithmic-depth quantum carry-lookahead adder, arXiv preprint quant-ph/0406142 (2004).
[69] Y. Takahashi, S. Tani, and N. Kunihiro, Quantum addition circuits and unbounded fan-out, arXiv preprint arXiv:0910.2530 (2009).
[70] M. K. Bhaskar, S. Hadfield, A. Papageorgiou, and I. Petras, Quantum algorithms and circuits for scientific computing, arXiv preprint arXiv:1511.08253 (2015).
[71] T. Albash and D. A. Lidar, Adiabatic quantum computation, Reviews of Modern Physics 90, 015002 (2018).
[72] F. Casas, A. Murua, and M. Nadnic, Efficient computation of the zassenhaus formula, Computer Physics Communications 183, 2386 (2012).
[73] S. Olver and A. Townsend, Fast inverse transform sampling in one and two dimensions, arXiv preprint arXiv:1307.1223 (2013).
Appendix

The structure of this appendix is as follows. We first introduce basic definitions and notation in Appendix A. In Appendix B we review relevant Hamiltonian simulation techniques of crucial importance which form the basis of our state-preparation procedure. These include 1-sparse in Appendix B.1 and low-rank in Appendix B.2 simulation techniques as well as adiabatic simulation in Appendix B.3.

We detail our adiabatic state-preparation approach in Appendix C. First we state a compact form of our main result in Theorem 2 and then describe details of the procedure in Appendix C.1. We then systematically derive our upper bounds, first for an arbitrary finite qubit count in Appendix C.2 and then we prove existence of constant asymptotic bounds in Appendix C.3 for continuous functions. We then outline generalisations to arbitrary functions (mappings) in Appendix C.4.

In the next three sections we detail further applications beyond our main result: we propose a ‘self-verifying’ state-preparation approach based on a phase estimation protocol in Appendix D as well as a more compact variant of it based on a Hadamard test in Appendix E. We also discuss that these two variants can be used to efficiently integrate functions while we point out the phase-estimation approach has a superior convergence rate. We finally discuss generalisation to continuous multi-variate functions. The last section Appendix H contains further technical details.

Appendix A: Preliminaries and definitions

1. Continuous and efficiently computable functions

While our approach naturally applies to arbitrary functions (mappings) as we detail in Appendix C.4, we mostly focus on the pivotal case of continuous functions given these contain most practically important cases and they naturally admit convenient asymptotic properties as we show later.

In this work we refer to a continuous function \( f : [a, b] \to \mathbb{C} \) on the bounded interval \( a \leq x \leq b \) as \( f(x) \) whose general definition is \( f \in \mathcal{C} \) as \( \lim_{c \to x} f(c) = f(x) \). Recall that due to the boundedness theorem all such functions are bounded as \( |f(x)| \leq \|f\|_{\text{max}} \). While we keep our results as general as possible and consider arbitrary continuous functions (unless stated otherwise), we note that so-called Lipschitz continuous functions cover nearly all instances one can encounter in practice, and we detail in Appendix H.2 that these Lipschitz continuous functions additionally satisfy powerful exponential scaling properties.

In the rest of this work we will refer to continuous functions and their discretisations interchangeably: in a digital (quantum) computer we aim to represent both the argument \( x \) and the function value \( f(x) \) in a binary encoding. While we explicitly discuss our discretisations below, let us now define a subset of general continuous functions which additionally satisfy efficient computability constraints with respect to some suitable binary encoding.

Definition 1 (efficiently computable functions). Given the above families of continuous functions, we consider \( f \) to be efficiently computable if there exists an efficient quantum algorithm \( O_f \) performing the mapping,

\[
O_f |x⟩|0⟩ = |x⟩|f(x)⟩,
\]

(A1)

for any computational basis state \( |x⟩ \) given a suitable binary encoding of the real and complex numbers \( x \) and \( f(x) \), respectively. Such an oracle may be efficiently constructed for any function where an efficient classical circuit allows the evaluation of \( f \) for any possible input in the domain. Intuitively, if there is a classical algorithm allowing the evaluation of \( f \), then it is possible to construct \( O_f \).

Let us consider a simple example. Consider the quantum circuit \( O_f \) implementing the function \( f(x) = x \). This can be implemented by a constant-depth circuit as a sequence of CNOT gates that are controlled on the individual bits in the \( |x⟩ \) register. Suppose the first register has \( n \) bits and the second has \( d \), if \( n > d \), we can simply just control on the first \( n \) bits of the first register. Furthermore, more involved functions can be computed efficiently by invoking addition and other arithmetic operations; these can be implemented as discussed in refs 50–54.

A large number of applications require that a quantum state encodes a probability distribution, i.e., a non-negative function. Given a non-negative function, its integral over any domain is also non-negative and we will see later that the ability to efficiently integrate a non-negative function greatly simplifies our general problem, i.e., through integration we effectively broaden possibly narrow peaks.
Definition 2 (efficiently integrable functions). We define the non-negative function \( g : [a, b] \to \mathbb{R}_+ \) such that \( g(x_j) \) gives the density in the grid interval between state \( |x_j\rangle \) and \( |x_{j+1}\rangle \),

\[
g(x_j) = \left[ \int_{x_j}^{x_j + \Delta N} f(x) \, dx \right]^{1/2},
\]

where \( f : [a, b] \to \mathbb{R}_+ \) is a non-negative function and we assume that the resulting integral function \( g(x_j) \) is efficiently computable as in Definition 1. Here, \( \Delta N \equiv \frac{b-a}{N} \). As such, we can efficiently enforce the normalisation \( \sum_j |g(x_j)|^2 = \|f\|_1 = 1 \) which we will assume is always the case.

Let us consider the function \( f(x) = x \) for example. We can analytically compute its integral as \( g(x_j) = \sqrt{\Delta N (\Delta N/2 + x_j)} \) and implement the corresponding oracle by using arithmetics operations.

Definition 3 (filling ratio). Given the usual definition for \( L^p \) function norms of a Riemann integrable, bounded function \( f \), \( \|f\|_p = \left[ \int_a^b |f(x)|^p \, dx \right]^{1/p} \), we define the filling ratio of \( f \),

\[
\mathcal{F} := \frac{\|f\|_1}{\|f\|_{\max}}.
\]

Note that for continuous functions over a closed interval \( \|f\|_\infty =: \|f\|_{\max} \) represents the absolute largest value of the function.

Above the filling ratio \( 0 < \mathcal{F} \leq 1 \) actually quantifies the absolute area under the function relative to a rectangle of the same maximal height; the upper bound (1) is saturated by a maximum entropy uniform function \( f(x) = \text{const} \) for which our presented techniques are most efficient. On the other hand, the filling ratio \( \mathcal{F} \) can be arbitrarily small but is never zero – the lower bound is approached by functions that consist of arbitrarily narrow peaks.

2. Encoding functions into quantum states

We are generally concerned with the preparation of quantum states following continuous functions. We now introduce some of the notation used throughout the remainder of this paper. First, we assume that we have an \( n \) qubit system, with \( N = 2^n \) computational basis states. Second, as explained above we consider continuous functions \( f(x) \) on some real interval \( x \in [a, b] \) and we discretise the interval into \( N \) equally spaced samples with \( a = x_0 \) is the leftmost value in the interval, and \( x_{N-1} \) is the rightmost value via \( b = x_N \). The step size is thus given by \( \Delta N = (x_N - x_0)/N = (b-a)/N \). Note that the binary integers \( j \in \{0, 1\}^n \) index the standard standard basis states \( \{|j\rangle\} \) of our quantum computer and we can equivalently refer to these in terms of the \( x \) values \( \{|x_j\rangle\} \) given the relation \( x_j = x_0 + j \Delta N \) admits the inverse mapping \( j = (x_j - x_0) \Delta N^{-1} \).

Suppose we have an \( n \) qubit quantum system, prepared in the state \( |\psi\rangle \). We define two kinds of encodings as

\[
\begin{align*}
\text{pointwise: } |\psi\rangle &= (\mathcal{N}_N)^{-1} \sum_{j \in \{0, 1\}^n} f(x_j) |j\rangle, \\
\text{integral: } |\psi\rangle &= \sum_{j \in \{0, 1\}^n} g(x_j) |j\rangle,
\end{align*}
\]

where the integral encoding is identical to that in the Grover-Rudolph approach detailed in Appendix H. Here we assume that \( f(x) \) is an efficiently computable function from Definition 2 and \( \mathcal{N}_N := \sum_{j \in \{0, 1\}^n} |f(x_j)|^2 \) is a normalisation constant which we cannot necessarily compute efficiently. Here \( g(x_j) \) are piecewise integrals of an efficiently integrable function from Definition 2 and, as such, we can efficiently enforce normalisation.

While the two encodings may have very different behaviours for finite \( N \) we show in Property 1 that for large \( N \) asymptotically the integral encoding becomes identical to the corresponding pointwise encoding of a non-negative function.

Remark 1. While in the present work we restrict ourselves to a standard, uniform grid representation as per Eq. (A3), it is straightforward to implement non-uniform grids that are required for, e.g., computing quadrature integrals. One would then proceed by efficiently computing \( f(q(x_j)) \) where \( q(x_j) \) efficiently maps the uniform grid to a non-uniform one and thus the composite function is efficiently computable.

Note that later we will also make use of a binary encoding of the function values \( f(x) \), which we we discuss in detail in Sec. [B 1].
Appendix B: Hamiltonian Simulation Techniques

1. 1-Sparse Hamiltonian Simulation

Let us now recollect the approach described in refs. \cite{23,26} for efficiently simulating the time evolution under 1-sparse Hamiltonians.

**Lemma 1.** Suppose we are given a one-sparse Hermitian Hamiltonian $H$ acting on $n$ qubits. $H$ is specified by the oracles $O_f$ and $O_H$, defined such that

\[
O_f|x_n|0_n = |x_n|f(x)_n, \quad (B1)
\]
\[
O_H|x_n|y_n|0_d = |x_n|y_n|H_{x,y}d. \quad (B2)
\]

where the function $f(x)$ gives the column of the only non-zero matrix element of $A$ in row $x$, $H_{x,y}$ corresponds to the element in the $x^{th}$ row and $y^{th}$ column of $H$, and we use $d$-bits of resolution to encode the matrix elements of $H$. Then, an initial state $|\psi_0\rangle$ can be simulated by $e^{-iHt}$ for an arbitrary $t$ with 0 error using $O(1)$ oracle calls (and circuit depth equal to the circuit depth of the oracle implementations).

**Proof.** As we include this proof simply to help the reader build intuition, we only consider the case where the matrix elements of $H$ are real, and defer the reader to the existing literature to see how this technique is adapted for the general case of a Hermitian $H$ \cite{23,26} (noting that the overall asymptotic complexity remains the same) \cite{23}. First, observe that for a one-sparse $H$ satisfying $H = H^*$, $H|x\rangle = |f(x)\rangle$ as $H$ maps $|x\rangle$ to the only non-zero row, which must be the same as the non-zero column in row $x$ since $H$ is symmetric. Moreover, it must also be the case that $f(f(x)) = x$, otherwise $H$ would not be one-sparse. As a result, $H$ can be thought of as an adjacency matrix describing a graph with $2^n$ vertices, such that the graph consists of a set of connected components with each component containing either one or two vertices. Therefore, $H$ acts on $1 \times 1$ and $2 \times 2$ invariant subspaces. Consequently, simulation of $e^{-iHt}$ is equivalent to simply performing simulations in each of these $1 \times 1$ and $2 \times 2$ subspaces. Thus, our simulation circuit needs the logic to identify if a given basis vector is in a $1 \times 1$ or a $2 \times 2$ block, and then needs to perform the corresponding simulation.

First, we write our initial quantum state in the general form $|\psi_0\rangle = \sum x=0 ^{2^n-1} \alpha_x|x\rangle$. We then initialize our quantum state coupled to an $n$-qubit ancillary register, a single qubit ancillary register, a $d$-qubit ancillary register, and a final 1-qubit ancillary register. Note that a more efficient implementation of this procedure exists, but we use the following for expository purposes. Our joint quantum state is thus given by,

\[
|\psi_0\rangle_n|0_n|0_1|0_d|1 = \sum x=0 ^{2^n-1} \alpha_x|x\rangle_n|0_n|0_1|0_d|1. \quad (B3)
\]

We begin the simulation procedure by applying the $O_f$ oracle to the first and second registers,

\[
|\psi_0\rangle_n|0_n|0_1|0_d|1 \xrightarrow{O_f} \sum x=0 ^{2^n-1} \alpha_x|x\rangle_n|f(x)_n|0_1|0_d|1. \quad (B4)
\]

We now define the comparison operator, CMP, which acts on the first three registers with the mapping $\text{CMP}|x\rangle_n|y\rangle_n|0_1 = |x\rangle_n|y\rangle_n|x > y\rangle_1$ (where $x > y$ is 1 if true, and 0 if false). Applying CMP yields,

\[
\text{CMP} \sum x=0 ^{2^n-1} \begin{cases} 
\alpha_x|x\rangle_n|f(x)_n|0_1|0_d|1 & \text{if } x \leq f(x) \\
\alpha_x|x\rangle_n|f(x)_n|1_1|0_d|1 & \text{if } x > f(x).
\end{cases} \quad (B5)
\]

We now define the controlled-swap operation, CSWAP, acting on the first three registers such that $\text{CSWAP}|x\rangle_n|y\rangle_n|0_1 = |x\rangle_n|y\rangle_n|0_1$ and $\text{CSWAP}|x\rangle_n|y\rangle_n|1_1 = |y\rangle_n|x\rangle_n|1_1$. Applying CSWAP yields,

\[
\text{CSWAP} \sum x=0 ^{2^n-1} \begin{cases} 
\alpha_x|x\rangle_n|f(x)_n|0_1|0_d|1 & \text{if } x \leq f(x) \\
\alpha_x|f(x)_n|x\rangle_n|1_1|0_d|1 & \text{if } x > f(x).
\end{cases} \quad (B6)
\]

---

1 Nathan Wiebe has an excellent video lecture covering 1-sparse Hamiltonian simulation in \cite{link}.
Thus, we can clearly implement this gate by a sequence of the binary expansion $d$
and apply CRz gate conditioned on the third register as their target input, we can
in a $2^{d}$ invariant subspace, and so the correct evolution follows a rotation by CRx. Noting that both the CRx and
CRz gates use the fourth register as their “pass-through” input, and the third register as their target input, we can
create the CRz gate (although it can trivially be created from the CRx gate by applying a Hadamard before and after
the gate on the target qubit).

We now must determine whether a given state is in a $1 \times 1$ or $2 \times 2$ invariant subspace, to do this we use the final ancillary qubit (note that this qubit is not necessary, and this operation can be done in place, however we include it for
notational clarity). Define a new operator, EQ, which acts on the first two registers and the fifth register, with
the mapping $E Q | x , y \rangle _ { n , f } | 0 \rangle _ { 1 } = | x , y \rangle _ { n , f } | x = y \rangle _ { n }$, where $x = y$ is 1 if true, and 0 otherwise. Then,

$$e^{−iαtX} = e^{−i(α_{1}2^{-d−1}α_{2}2^{-d−2}+\ldots+α_{d}2^{-d−d})tX} = e^{−iα_{1}2^{-d−1}tX} e^{−iα_{2}2^{d−2}tX} \ldots e^{−iα_{d}2^{0}tX}. \quad (B8)$$

Thus, we can clearly implement this gate by a sequence of $d$ single-qubit controlled Rx gates, with
$R x (t) = \cos(\frac{t}{2}) I − i \sin(\frac{t}{2}) X$, where the $j$th gate is controlled on bit $a_{j}$ and the time is set to $t2^{d−j}$. Thus, the cost of this operation scales as $O(d)$ in terms of circuit depth and required parameterized gates. The same argument applies to creating the
CRz gate (although it can trivially be created from the CRx gate by applying a Hadamard before and after
the gate on the target qubit).

We now must determine whether a given state is in a $1 \times 1$ or $2 \times 2$ invariant subspace, to do this we use the final ancillary qubit (note that this qubit is not necessary, and this operation can be done in place, however we include it for
notational clarity). Define a new operator, EQ, which acts on the first two registers and the fifth register, with
the mapping $E Q | x , y \rangle _ { n , f } | 0 \rangle _ { 1 } = | x , y \rangle _ { n , f } | x = y \rangle _ { n }$, where $x = y$ is 1 if true, and 0 otherwise. Then,

$$e^{−iαtX} = e^{−i(α_{1}2^{-d−1}α_{2}2^{-d−2}+\ldots+α_{d}2^{-d−d})tX} = e^{−iα_{1}2^{-d−1}tX} e^{−iα_{2}2^{d−2}tX} \ldots e^{−iα_{d}2^{0}tX}. \quad (B8)$$

Thus, we can clearly implement this gate by a sequence of $d$ single-qubit controlled Rx gates, with
$R x (t) = \cos(\frac{t}{2}) I − i \sin(\frac{t}{2}) X$, where the $j$th gate is controlled on bit $a_{j}$ and the time is set to $t2^{d−j}$. Thus, the cost of this operation scales as $O(d)$ in terms of circuit depth and required parameterized gates. The same argument applies to creating the
CRz gate (although it can trivially be created from the CRx gate by applying a Hadamard before and after
the gate on the target qubit).

We now must determine whether a given state is in a $1 \times 1$ or $2 \times 2$ invariant subspace, to do this we use the final ancillary qubit (note that this qubit is not necessary, and this operation can be done in place, however we include it for
notational clarity). Define a new operator, EQ, which acts on the first two registers and the fifth register, with
the mapping $E Q | x , y \rangle _ { n , f } | 0 \rangle _ { 1 } = | x , y \rangle _ { n , f } | x = y \rangle _ { n }$, where $x = y$ is 1 if true, and 0 otherwise. Then,

$$e^{−iαtX} = e^{−i(α_{1}2^{-d−1}α_{2}2^{-d−2}+\ldots+α_{d}2^{-d−d})tX} = e^{−iα_{1}2^{-d−1}tX} e^{−iα_{2}2^{d−2}tX} \ldots e^{−iα_{d}2^{0}tX}. \quad (B8)$$

Thus, we can clearly implement this gate by a sequence of $d$ single-qubit controlled Rx gates, with
$R x (t) = \cos(\frac{t}{2}) I − i \sin(\frac{t}{2}) X$, where the $j$th gate is controlled on bit $a_{j}$ and the time is set to $t2^{d−j}$. Thus, the cost of this operation scales as $O(d)$ in terms of circuit depth and required parameterized gates. The same argument applies to creating the
CRz gate (although it can trivially be created from the CRx gate by applying a Hadamard before and after
the gate on the target qubit).

We now must determine whether a given state is in a $1 \times 1$ or $2 \times 2$ invariant subspace, to do this we use the final ancillary qubit (note that this qubit is not necessary, and this operation can be done in place, however we include it for
notational clarity). Define a new operator, EQ, which acts on the first two registers and the fifth register, with
the mapping $E Q | x , y \rangle _ { n , f } | 0 \rangle _ { 1 } = | x , y \rangle _ { n , f } | x = y \rangle _ { n }$, where $x = y$ is 1 if true, and 0 otherwise. Then,
where the second last equation follows from the fact that if \( x < f(x) \), then \( |x\rangle_n|f(x)\rangle_n \) is equal to \( |f(x)\rangle_n|x\rangle_n \) when \( f(x) > x \), and the notation \( \sum_{x=f(x)} \) means sum over all \( x \) such that \( x = f(x) \), and similarly \( \sum_{x< f(x)} \) means sum over all \( x \) such that \( x < f(x) \). Finally, we uncompute all the ancillary operations (EQ, \( O_H \), CSWAP, CMP, \( O_f \)), obtaining the state

\[
\sum_{x=f(x)} e^{-iH_{x,s}t} \alpha_{x} |x\rangle_n |0\rangle_n |0\rangle_1 |0\rangle_d |0\rangle_1 + \sum_{x\neq f(x)} \alpha_{x} \left( e^{-iH_{x,t}t} |x\rangle_n \right) |0\rangle_n |0\rangle_1 |0\rangle_d |0\rangle_1 \tag{B15}
\]

\[
= \sum_{x=0}^{2^n-1} \alpha_{x} \left( e^{-iH_{x,t}t} |x\rangle_n \right) |0\rangle_n |0\rangle_1 |0\rangle_d |0\rangle_1 = \left( e^{-iH_{x,t}t} |\psi_0\rangle_n \right) |0\rangle_n |0\rangle_1 |0\rangle_d |0\rangle_1. \tag{B16}
\]

where this simply follows from the observation that \( e^{-iH_{x,s}t} \) correctly describes the evolution of a \( 1 \times 1 \) invariant subspace, while \( e^{-iH_{x,t}t} \) correctly describes the evolution of a \( 2 \times 2 \) invariant subspace, and then we simply apply linearity of \( e^{-iH_{t}} \).

It is worth briefly mentioning that if the matrix elements of \( H \) can be exactly encoded in \( d \) bits, then this procedure incurs 0 error, and if they cannot, then the error vanishes exponentially in \( d \) and so is effectively negligible. \( \square \)

Note that if the elements of \( H \) are efficiently computable, then a quantum circuit implementing \( O_H \) can clearly be implemented efficiently (i.e. with polynomial bounded complexity). Similarly, if the location of the non-zero element in any given row can also be computed efficiently, then the overall one-sparse simulation circuit can also be implemented efficiently.

2. Low-rank Hamiltonian Simulation

As low-rank Hamiltonian simulation is an essential subroutine in our algorithm, we now recollect results of Rebentrost et al. and summarize their proof \[22\].

Statement 1. The procedure presented in \[22\] allows the dynamics under an arbitrary low-rank dense matrix \( A \in \mathbb{C}^{n \times n} \) to be simulated efficiently if the matrix satisfies the two conditions: \( \|A\|_{\text{max}} = \Theta(1) \) and \( \|A\|_2 = \Theta(N) \) as we increase the dimension \( N \). Moreover, we suppose we have oracular access to the elements of \( A \). Then, for time \( \Delta t \), we can simulate the dynamics under \( e^{-i\Delta t A} \) using 4 oracle queries, with trace-norm error bounded by \( 2\|A\|_{\text{max}}^2 \Delta t^2 \).

Proof. We now summarize the results of \[22\]. First, we begin by embedding the elements of \( A \) in a matrix \( S_A \in \mathbb{C}^{N^2 \times N^2} \) as follows,

\[
S_A = \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} A_{jk} |k\rangle \langle j| \otimes |j\rangle \langle k|. \tag{B17}
\]

Clearly, \( S_A \) is one-sparse. Moreover, element \( x, y \) of \( S_A \), \( \langle S_A \rangle_{xy} \) can be computed easily and is given by \( A_{jk} \), where \( x = jN + k \) and \( y = jN + k \). Thus, \( O_{S_A} \) is clearly efficiently implementable, given access to the elements of \( A \). Moreover, given the structure of \( S_A \) it is also straightforward to implement \( O_f \), and as a result we can perform the simulation \( e^{-iS_A \Delta t} \) using Lemma \[1\] with a total of 4 oracle calls, and with 0 error (assuming no discretization error in the matrix elements of \( A \)). We now assume that the state we wish to evolve has a density operator given by \( \sigma \), and we have a separate \( n \)-qubit register in state \( \rho = \langle \phi| \langle \phi \rangle \) such that \( \langle \phi \rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle \) (i.e. a uniform superposition). The exact evolution for time \( \Delta t \) under \( S_A \) of state \( \rho \otimes \sigma \) generates the following dynamics under the partial trace,

\[
\text{tr}_1[e^{-i\Delta t S_A \rho} \otimes \sigma e^{i\Delta t S_A}] = e^{-i\Delta t A/N} \sigma e^{i\Delta t A/N} + \epsilon_0. \tag{B18}
\]

The leading error in \( \Delta t \) in terms of a trace distance was established in \[22\] as

\[
\epsilon_0 \leq 2\|A\|_{\text{max}} \Delta t^2, \tag{B19}
\]

which depends on the absolute largest entry of the matrix \( \|A\|_{\text{max}} \). We will use this approach to simulate the dynamics under our rank-1 matrix as \( A/N \propto |\psi\rangle \langle \psi| \). \( \square \)
3. Adiabatic Computation

Given an initial Hamiltonian $H(0) = H_0$ whose ground state is easy to prepare, and a final Hamiltonian $H(1) = H_1$ whose ground state we would like to prepare, we define the time-dependent Hamiltonian $H(s)$, with $0 \leq s \leq 1$. If we evolve the state according to $H(s)$ “sufficiently slowly”, the adiabatic theorem guarantees that the state will remain in the instantaneous ground state of $H(s)$ throughout the evolution, and so at the end of the procedure we will have obtained the ground state of $H_1$. For an insightful discussion of adiabatic quantum computation, see the paper by van Dam et al. [32], or the comprehensive review by Albash et al. [33]. As introduced by Farhi et al. [34], $H(s)$ is often defined as an interpolation between $H_0$ and $H_1$ in the form $H(s) = (1 - s)H_0 + sH_1$, however, this generally need not be the case. Precisely, we perform the evolution obtained by solving the Shrödinger equation,

$$\frac{d}{ds}|\psi(s)\rangle = -iH(s)|\psi(s)\rangle,$$

where we select $|\psi(s)\rangle$ such that $|\psi(0)\rangle$ is the ground state of $H_0$. Then, our objective is to obtain a circuit performing the evolution under the time-dependent Hamiltonian $H(s)$. Of note, the adiabatic evolution is considered sufficiently slow if throughout the evolution the adiabatic schedule $\tau(s)$ satisfies,

$$\frac{\|\frac{d}{ds}H(s)\|_2}{g^2(s)} \ll \tau(s),$$

where $g(s)$ is the spectral gap of $H(s)$ (i.e. the difference in the two lowest energy levels), and $\|\frac{d}{ds}H(s)\|_2$ is the spectral norm of the derivative of our time-dependent Hamiltonian. The spectral norm of a matrix $M$ is defined as

$$\max_{\|x\|_2=1} \|Mx\|_2.$$  

In the simplest implementation it is sufficient to use a constant schedule throughout, defined as

$$\frac{\max_{s \in [0,1]} \|\frac{d}{ds}H(s)\|_2}{g_{\min}} \ll \tau$$

where $g_{\min}$ is the minimum spectral gap of $H(s)$ for any time $s$. Then, for the constant schedule, the delay factor is given by $T = \int_0^1 \tau ds = \tau$ and so we have the bound $T = \mathcal{O}\left( \max_{s \in [0,1]} \frac{\|\frac{d}{ds}H(s)\|_2}{g_{\min}} \right)$. As described by van Dam et al., when using a constant schedule, a discretized unitary transformation induced by $H(t)$ may be written as,

$$U'(T) = \exp\left[ -i\frac{T}{r} H(\frac{r}{r}) \right] \cdots \exp\left[ -i\frac{T}{r} H(\frac{1}{r}) \right].$$

That is, we approximate the overall transformation by a sequence of $r$ time-independent evolutions, each applied for an equal amount of time $\frac{T}{r}$. An important fact we utilize in bounding the error from such a discretization comes from Lemma 1 presented by van Dam et al. [32]. We summarize their lemma and describe its proof for completeness.

**Lemma 2.** Suppose you have two time-dependent Hamiltonians, $H(t)$ and $H'(t)$ such that they induce the unitaries $U(T)$ and $U'(T)$ respectively. If the spectral norm of the difference of the Hamiltonians is bounded by at most $\delta$ for all times $t$, i.e. $\|H(t) - H'(t)\|_2 \leq \delta$, then $\|U(T) - U'(T)\|_2 \leq \sqrt{2T\delta}$.

**Proof.** Define the states $|\psi'(t)\rangle$ and $|\psi(t)\rangle$, such that $|\psi'(0)\rangle = |\psi(0)\rangle$ which are respectively obtained by evolving the starting state according to $H'(t)$ and $H(t)$. We then start by computing the operator distance via the (maximal) vector norm $\|(U(T) - U'(T))|\psi(0)\rangle\|_2 = \sqrt{\langle \psi(0)| (U(T) - U'(T))^\dagger (U(T) - U'(T)) |\psi(0)\rangle}$, we have

$$\|U(T) - U'(T)\|_2 \leq \sqrt{2 - 2\Re\langle \psi'(T)|\psi(T)\rangle}.$$  

Here we consider how the inner product of the two states change with respect to time (and note the parametrisation is smooth in our case), obtaining

$$\frac{d}{dt} \langle \psi'(t)|\psi(t)\rangle = -i\langle \psi'(t)|(H(t) - H'(t))|\psi(t)\rangle.$$  

We can then integrate both sides with respect to time from 0 to $T$ and lower bound its absolute value as

$$|\langle \psi'(T)|\psi(T)\rangle| = |1 - i \int_0^T \langle \psi'(t)|(H(t) - H'(t))|\psi(t)\rangle \ dt| \geq 1 - i \int_0^T \langle \psi'(t)|(H(t) - H'(t))|\psi(t)\rangle \ dt|$$

$$
\text{(B20)}
$$
δH = t

δH = δ/t

|δH−δ/t|

\[ \delta \parallel \text{distance of two unitaries} \]

Distance is bounded by \( \delta \parallel \text{distance} \) of two unitaries. By assumption the spectral norm of the difference of the two Hamiltonians is bounded by \( \delta \parallel \text{distance} \) of two unitaries, \( \delta = \delta \parallel \text{distance} \) of two unitaries.

Corollary 1. We now take the power series expansion of each exponential, keeping only terms of order \( t^3 \) and below, the effective Hamiltonian \( H'' = e^{-itH''} \) that generates the time-independent Hamiltonians \( H' \) and \( H'' \) respectively, we aim to bound the distance \( \delta_H := \|H' - H''\|_2 \). We can express this operator distance in terms of the (maximum of the) vector norm

\[ \|U' - U''\|_2 = \langle \psi|(U' - U'')(U' - U'')^\dagger|\psi\rangle = 2 - 2\Re(\langle \psi|U'U'|\psi\rangle). \]  

(B26)

We now use the Baker-Campbell-Hausdorff formula, in the form of the Zassenhaus formula \( e^{tX}e^{tY} = e^{tX}e^{-\frac{t^2}{2}[X,Y]}e^{\frac{t^3}{3!}Q} \cdots \) (with \( Q = 2[Y,X,Y] + [X,X,Y] \)), which we can rearrange to obtain

\[ e^{tX}e^{tY} = e^{t(X+Y)} \cdots e^{-\frac{t^3}{3!}Q}e^{\frac{t^2}{2}[X,Y]}. \]

We apply this formula by substituting \( X := iH' \) and \( Y := -iH'' \) as

\[ (U')^tU'' = e^{itH'}e^{-itH''} = e^{it(H' - H'')} \cdots e^{-\frac{t^3}{3!}Q}e^{\frac{t^2}{2}[H',H'']}. \]

We now take the power series expansion of each exponential, keeping only terms of order \( t^3 \) and below,

\[ (U')^tU'' = \left( 1 + it(H' - H'') - \frac{t^2}{2}(H' - H'')^2 - i\frac{t^3}{3!}(H' - H'')^3 \right) \left( 1 - \frac{t^2}{2}Q \right) \left( 1 + t^2[H', H''] \right) + O(t^4) \]

\[ = 1 + it(H' - H'') + \frac{t^2}{2} \left( [H', H''] - (H' - H'')^2 \right) + t^3 \Xi + O(t^4), \]

where we have denoted the operator with factor \( t^3 \) as

\[ \Xi := -\frac{1}{6}Q + i\frac{t^2}{2}[H', H''] - i\frac{t^3}{3!}(H' - H'')^3. \]

We now want to compute the expected value term from Eq. \( B26 \) and obtain

\[ \Re(\langle \psi|U'U''|\psi\rangle) = 1 - \frac{t^2}{2}\Re(\langle \psi|(H' - H'')^2|\psi\rangle) + t^3\Re(\langle \psi|\Xi|\psi\rangle) + O(t^4), \]  

(B27)
using that the expected value of any Hermitian operator is real and thus \( \text{Re}\ i\langle \psi | (H' - H'') | \psi \rangle = 0 \), and similarly the expected value of any anti-Hermitian operator is imaginary and thus \( \text{Re}\langle \psi | [H', H''] | \psi \rangle = 0 \). Furthermore, rather than explicitly computing the value of the third-order term, we simply bound it as \( |\text{Re}\langle \psi | \Xi | \psi \rangle| \in \mathcal{O}(\delta_H) \). Therefore, we obtain

\[
\text{Re}\langle \psi | (U')^\dagger U'' | \psi \rangle = 1 - \frac{t^2}{2} \langle \psi | (H' - H'')^2 | \psi \rangle + O(t^3 \delta_H) + O(t^4) = 1 - \frac{t^2}{2} \|(H' - H'')| \psi \rangle \|^2 + O(t^3 \delta_H) + O(t^4).
\]

We substitute this back and obtain our bound as

\[
\|U' - U''| \psi \rangle\|^2 = t^2 \|(H' - H'')| \psi \rangle \|^2 + O(t^3 \delta_H) + O(t^4)
\]

where we now divide by \( t^2 \) and take the square root

\[
\|U' - U''| \psi \rangle\|/t = \sqrt{\|(H' - H'')| \psi \rangle \|^2 + O(t \delta_H) + O(t^2)}
\]

which can be computed using the expansion \( \sqrt{a + \epsilon} = \sqrt{a} + \mathcal{O}(\epsilon) \) as

\[
\|U' - U''| \psi \rangle\|/t = \|(H' - H'')| \psi \rangle + O(t \delta_H) + O(t^2).
\]

Given that this holds for any state \( |\psi\rangle \), the above equation implies the equivalence of the norms as

\[
\delta_H = \delta/t + O(t \delta_H) + O(t^2),
\]

where we have substituted our expressions for the norms as \( \delta \equiv \|U' - U''\|_2 \) and \( \delta_H \equiv \|H' - H''\|_2 \). Furthermore, we conclude by re-expressing the additive error as a multiplicative error \( \delta_H + O(t \delta_H) = \delta_H (1 + O(t)) \) as

\[
\delta_H = \left( \delta/t + O(t^2) \right) \left( 1 + O(t) \right) = \delta/t + \delta + O(t^2) \leq \kappa t + O(t^2),
\]

where in the last equation we substituted our upper bound on the unitary distance as \( \delta \leq \kappa t^2 \).

**Remark 2.** Corollary 1 bounds the maximum distance of the Hamiltonians generating unitaries with bounded distance. However, this bound only holds for small \( t < 1 \), and such a general statement is not possible for non asymptotic small \( t \), as the following counterexample proves. Suppose we have Hamiltonians \( H' = \text{diag}(1, \lambda_2, \ldots, \lambda_d) \) and \( H'' = \text{diag}(2, \lambda_2, \ldots, \lambda_d) \) which share the same eigenvalues except for one eigenvalue and thus the distance is \( \|H' - H''\|_2 = 1 \). However, the induced unitaries at \( t = 2\pi \) are identical as \( U' = \text{diag}(1, e^{-i2\pi \lambda_2}, \ldots, e^{-i2\pi \lambda_d}) = U'' \).

**Lemma 3.** Suppose you have Hamiltonians \( \mathcal{H}(t), \mathcal{H}'(t) \) and \( \mathcal{H}''(t) \), which induce the unitary transformations \( U(T), U'(T) \) and \( U''(T) \) respectively, and that \( \|\mathcal{H}(t) - \mathcal{H}'(t)\|_2 \leq \delta_0 \) and \( \|\mathcal{H}'(t) - \mathcal{H}''(t)\|_2 \leq \delta_1 \). Then, \( \|U(T) - U''(T)\|_2 \leq \sqrt{2T(\delta_0 + \delta_1)} \).

**Proof.** First, by the triangle inequality,

\[
\|\mathcal{H}(t) - \mathcal{H}''(t)\|_2 = \|\mathcal{H}(t) - \mathcal{H}'(t) + \mathcal{H}'(t) - \mathcal{H}''(t)\|_2 \leq \|\mathcal{H}(t) - \mathcal{H}'(t)\|_2 + \|\mathcal{H}'(t) - \mathcal{H}''(t)\|_2 \leq \delta_0 + \delta_1.
\]

Applying Lemma 2 immediately then gives the result \( \|U(T) - U''(T)\|_2 \leq \sqrt{2T(\delta_0 + \delta_1)} \).

**Appendix C:** General distribution loading through adiabatic evolution: direct state preparation on \( n \) qubits

Let us now compactly present the main result of the present section only in the asymptotic limit \( N \to \infty \).

**Theorem 2.** Asymptotically in \( N \), for an efficiently computable continuous function \( f : [a, b] \to \mathbb{C} \), the direct state preparation algorithm prepares the \( n \) qubit state (with \( N = 2^n \), \( |\psi\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} f(x_j) |j\rangle \) (with \( f \) replaced with \( g \) in the case of the integral encoding, as defined in Definition 2), up to error \( \epsilon \) with query complexity bounded as

\[
O\left( \frac{FP}{\epsilon^2} \right),
\]

where \( F \) is the filling ratio from Definition 3 and \( p = -2 \) if using the integral function encoding, and \( p = -4 \) if using the point-wise function encoding. The error \( \epsilon \) is the deviation from an ideal unitary in terms of a spectral distance. Moreover, the algorithm uses \( O(n + d) \) ancillary qubits (where \( d \) is the number of digits used in the discretization of \( f \) ), and has a probability of failure bounded by

\[
O\left( \frac{\epsilon^2}{\epsilon^2} \right).
\]

Thus the probability of failure decreases as we increase the target precision \( \epsilon \), a feature that is highly advantageous.
Proof. From Lemma \[\text{[10]}\] we know that the algorithm has query complexity \(O(r)\), and uses \(O(n + d)\) ancillary qubits. From Corollary \[\text{[6]}\], we have that \(\epsilon \in O(\sqrt{F}/r)\), and so it immediately follows that to obtain an error \(\epsilon\) we must have \(r \in O(F^{1/2}/\epsilon^2)\), yielding the query bound. Finally, from Lemma \[\text{[2]}\] we have that the probability of failure is bounded by \(O(F^{1/2}/r)\), which immediately gives the probability failure bound of \(O(\epsilon^2)\) when substituting our above upper bound for \(r\) as a lower bound, i.e., we use \(r\) larger than the upper bound to guarantee a worst-case error \(\epsilon\). Similar statements can be made without assuming large \(N\) by directly combining the corresponding non-asymptotic lemmas from Appendix \[\text{[C2]}\]. \(\square\)

Note that in Appendix \[\text{[C2]}\] we will indeed derive explicit upper bounds for arbitrary finite qubit count \(n\) and above we have only stated the main result in the limiting scenario for brevity. Furthermore, we note that while the above asymptotic limits exist for any continuous functions, in the specific case of Lipschitz continuous functions (which cover nearly all scenarios in practice) these limits are approached rapidly in exponential order in \(n\). For commonly encountered non-pathological functions, the asymptotic limit is usually reached with \(n < 30\).

Remark 3. When implementing the approach in practice explicit knowledge of the filling ratio \(F\) is not actually required. The above result merely establishes the scaling of the number of timesteps \(r\) required to obtain an error \(\epsilon\), and note that in practice a bound on \(r\) is straight-forward to obtain by combining Lemma \[\text{[2]}\] and Remarks \[\text{[6]}\] and \[\text{[2]}\] as long as one can either bound or over-estimate the maximum value of \(f\) – which indeed bounds \(F\). We present the bounds in terms of \(F\) because it makes the intuition clear as to when the constant cost of the algorithm will be large or small.

Remark 4. In general, while not necessary, QRAM can provide an asymptotic improvement in the circuit depth by alleviating the need for most of the quantum arithmetic involved in implementing the oracle (e.g. by storing \(f_0\) and \(f_1\) for all values in the grid, and then using basic addition and multiplication to compute \(f_s\) as needed). Moreover, in certain QRAM architectures, (namely where each quantum register in the architecture has its own controlling classical mini-cpu and batch instructions can be instructed to all the classical controllers in parallel) the cost of the oracle access can be made \(O(1)\), thus rendering the algorithm’s circuit complexity the same as its query complexity.

1. The procedure

In the standard model of adiabatic quantum computation originally proposed by Farhi et al. in 2000 \[\text{[3]}\], one defines a time-dependent interpolated Hamiltonian \(H(s) = (1 - s)H_0 + sH_1\) where \(H_0\) is some initial Hamiltonian with an easy to prepare ground state, and \(H_1\) is a final Hamiltonian whose ground state we wish to prepare. One then evolves the initial state according to \(H(s)\) to obtain the final ground state of the problem. However, a challenge with this approach is that the spectral gap can become exponentially small in general (thus necessitating exponential evolution time).

However, in some cases this problem can be circumvented by viewing the paradigm of adiabatic quantum computation more generally – we need not limit ourselves to using a time-dependent Hamiltonian of the form \(H(s) = (1 - s)H_0 + sH_1\). In our case, we begin at \(n\) qubits (i.e. use a resolution of \(N = 2^n\)) and define a time-dependent rank-one Hamiltonian. That is, rather than interpolating between two separate Hamiltonians, we define our Hamiltonian as proportional to a projector onto a discretized time-dependent quantum state the encodes a parametrised function. Precisely, we define

\[
\mathcal{H}(s) := -A(s)/N \quad \text{pointwise:} \quad A(s) \text{ encodes } f_s \quad f_s := (1-s)f_0 + sf_1. \tag{C1}
\]

\[
\text{integral:} \quad A(s) \text{ encodes } g_s \quad g_s := (1-s)g_0 + sg_1.
\]

Here \(A(s)\) is a rank-one matrix with entries \([A(s)]_{kl} := f_s(x_k)f^*_s(x_l)\) (\([A(s)]_{kl} := Ng_s(x_k)g_s(x_l)\)). Furthermore, \(f_0(x)\) and \(g_0(x)\) are trivial functions, such as the uniform distribution, that we can easily prepare as our initial state while \(f_1(x)\) and \(g_1(x)\) are our final desired functions. This definition of \(\mathcal{H}(s)\) has numerous beneficial properties, which enable general analytic performance guarantees, which we now derive. As a matter of notation, we define the normalization constant \(\mathcal{N}(s)\) as

\[
\mathcal{N}(s) := \sqrt{\sum_{j=0}^{N-1} |f_s(x_j)|^2}, \tag{C2}
\]

for the interpolated function \(f_s\) (and similarly for the integration case \(g_s\)) as per Eq. \[\text{[C1]}\]. \(\mathcal{N}(1)\) is the main quantity we are concerned with in our proofs, and we derive all of our general bounds in terms of it (noting that as per Remark \[\text{[3]}\] it grows as \(O(\sqrt{N})\)).
In the case of the pointwise state preparation approach, while it is not a limitation of the presented algorithm, it is important to keep in mind that we define our problem such that we aim to represent a continuous function over a finite grid – which is only meaningful if the grid is fine enough to not miss features of the function. While we prove asymptotic properties for an increasing resolution $N \to \infty$, the approach might fail in ill-defined cases when $N$ is too small. In particular, if the function $f_1$ consists of only narrow peaks of width $\epsilon$ then we need at least $n \in O(\log_2(\epsilon^{-1}))$ qubits to resolve these peaks and for qubit counts below this threshold our discretisation may in a worst-case scenario correspond to the null-vector (i.e., the function value is 0 at all grid points). A particular example could be the function $f(x) = \sin(20\pi x)$ which requires at least 21 qubits of resolution in the range $0 \leq x \leq 1$ using our standard grid. It is important to stress that this is not a limitation of the algorithm being presented – when preparing a function according to point-wise samples no algorithm can handle such ill-defined cases. However, in practice this is unlikely to ever be a problem, because of the exponential growth of the grid in the number of qubits. For example, at 128 qubits, $N$ is sufficiently large to resolve a function defined on a kilometer interval with grid points at the Planck scale.

Let us now summarize the algorithm, along with the main proofs.

- We begin by preparing the state $|\psi_0\rangle = \frac{1}{\sqrt{N(1)}} \sum_{j=0}^{N-1} f_0(x_j)|j\rangle$, which by definition of $f_0$ is easy to prepare.
- We then implement adiabatic evolution for a total time $T$ according to the time-dependent rank-one Hamiltonian $\mathcal{H}(s)$ as defined in Lemma 4.
- We prove that $T$ is constant bounded in Lemma 5 (to see that this is indeed a constant bound, consider Remark 5).
- The adiabatic evolution is implemented by the sequence of time-independent unitary transformations given by $e^{-iT(1)} e^{-iT(2)} \cdots e^{-iT(1)}$.
- We prove a bound on the error incurred by this discretization of the adiabatic evolution in Lemma 6.
- In Lemmas 7 and 9 we demonstrate how each of the time-independent terms $e^{-iT(1)}$ may be implemented using the low-rank Hamiltonian simulation procedure, and bound the error and probability of failure incurred in the simulation.
- In Lemma 8 we bound the cumulative probability of failure from implementing the low-rank Hamiltonian simulation technique $r$ times in a row (in $e^{-iT(1)} e^{-iT(2)} \cdots e^{-iT(1)}$).
- In Lemma 11 we bound the total error incurred in the procedure.
- While we present these bounds for an arbitrary, finite qubit count $n$ in Appendix C 2 without assuming continuity of the functions, we establish asymptotic limits in Appendix C 3 using properties of continuous functions.
- Finally, in Appendix C 4 we establish that our approach is efficient not only for continuous but also for arbitrary functions (mappings) that satisfy certain properties.

2. Proofs for finite qubit count $n$

As we noted above we are primarily concerned with the normalisation factor from Eq. (C2) when proving performance properties of the adiabatic approach. Let us here point out that we can efficiently estimate this normalisation constant.

**Remark 5 (Efficiently computable normalisation).** As demonstrated in Appendices D 4 and E 2, it is possible to efficiently compute the normalization factor for an arbitrary $f_1$, and thus to re-scale $f_1$ such that $N(1) = \sqrt{N}$.

The above property will ensure that our following upper bounds on the performance of adiabatic evolution are simplified and tight. In the following subsection we first prove that the adiabatic evolution needs to run only for a constant bounded time $T$ by bounding the delay factor. The main results are illustrated in Fig. 6 (left). In the second subsection we prove bounds on the algorithmic errors incurred by our procedure. The algorithmic error of our adiabatic evolution is illustrated in Fig. 6 (right).
FIG. 6. Numerically verifying bounds that guarantee the efficient performance of adiabatic evolution from Appendix C 2 (left) The energy gap \( g(s) \) of our rank-1 Hamiltonian \( H(s) = -A(s)/N \) from Eq. (C1) is determined by the normalisation \( N(s) \) from Eq. (C2) and is bounded via Lemma 4. The matrix norm of the derivative \( \frac{dH(s)}{ds} \) relative to this spectral gap determines the overall evolution time \( T \) required for adiabatic evolution. Upper bound (dashed black line) from Lemma 4 (right) The matrix distance \( \| H(s) - H'(s) \|_2 \) determines the error when approximating a continuous adiabatic evolution via a piecewise constant evolution. We note that with similar proof techniques we could derive tighter asymptotic (in \( \Delta s \), not presented in this work) upper bounds (dashed red line) but we prefer the upper bound (dashed black line) in Lemma 6 due to its more compact expression. Given all bounds in Appendix C 2 are valid to arbitrary rank-1 matrices \( A \) (i.e., we need only invoke continuity of functions when computing asymptotic limits in Appendix C 3), we simulated 5000 unitary Haar-random discretisation vectors \( v_1 \) at \( n = 10 \) qubits for uniformly randomly selected \( 0 \leq s \leq 1 \) and logarithmically uniformly randomly selected vector norms \( N \).

### a. Proofs on boundedness of the spectral gap

**Lemma 4 (Spectral gap bound).** We choose the phase of our trivial initial function properly via the efficient verification protocol in Appendix E 3 as one of the four possibilities \( f_0(x) = (\pm 1 \pm i)/\sqrt{2} \). Given a parameterised rank-1 encoding \( A(s) \) from Eq. (C1), the spectral gap of our Hamiltonian \( H(s) = -A(s)/N \) is determined by the vector norm \( g(s) = |||v_s|||^2 \) and is bounded as

\[
g(s) \geq \frac{N(1)^2/N}{N(1)^2/N + 1}.
\]

**Proof.** First, observe that since \( H(s) \) is rank-one, its spectral gap is the same as its spectral norm, and thus

\[
g(s) = \| H(s) \|_2 = \left\| \frac{A(s)}{N} \right\|_2 = \| v_s \langle v_s \|_2 = \| v_s \|^2.
\]

Moreover, the definition for \( H(s) \) in Equation (C1) is equivalent to the following,

\[
\langle v_s \rangle = (1 - s)|v_0⟩ + s|v_1⟩,
\]

where \( |v_0⟩ = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} f_0(x_j)|j⟩ \) and \( |v_1⟩ = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} f_1(x_j)|j⟩ \) (and thus \( |v_0⟩, |v_1⟩ \) and \( |u_s⟩ \) are not normalized). In the integration case, simply replace each \( f(x) \) with the appropriate integral of \( f \) (as per Equation A3). Then, using the Equation (C4)

\[
g(s) = \| v_s \|^2 = (1 - s)^2|v_0⟩⟨v_0| + s^2|v_1⟩⟨v_1| + 2s(1 - s)\text{Re}\langle v_0|v_1⟩.
\]

Noting that \( \langle v_0|v_0⟩ = \frac{1}{N}N(0)^2 \), \( \langle v_1|v_1⟩ = \frac{1}{N}N(1)^2 \), and that by selecting the appropriate phase and sign in \( f_0(x) = (\pm 1 \pm i)/\sqrt{2} \) we ensure that \( 2\text{Re}\langle v_0|v_1⟩ \geq 0 \), we get

\[
g(s) \geq (1 - s)^2\frac{N(0)^2}{N} + s^2\frac{N(1)^2}{N}.
\]

Noting that it is trivial to select an \( f_0 \) (or \( g_0 \)) such that \( N(0)/\sqrt{N} = 1 \), the above simplifies to

\[
g(s) \geq (1 - s)^2 + s^2\frac{N(1)^2}{N}.
\]
Minimizing the expression with respect to \( s \), we find the minimum at \( s = \frac{1}{N(1)^2/N + 1} \). Substituting this value for \( s \), we then obtain the bound

\[
g(s) \geq \frac{N(1)^2}{N(1)^2 + N}.
\] (C8)

As an aside, \( 2\text{Re}(v_0|v_1) \geq 0 \) only holds if we have properly chosen the phase of our initial trivial function \( f_0(x) = (\pm 1 \pm i)/\sqrt{2} \). If we select the phase incorrectly, it is possible that \( \text{Re} \sum_{j} f_0(x_j) f_1(x_j) < 0 \), which negative term can decrease the spectral gap in Eq. (C5) below our lower bound (which assumes a non-negative \( \text{Re} \sum_{j} f_0(x_j) f_1(x_j) \)). Indeed for real functions we need only consider the sign \( f_0(x) = \pm 1 \) while for the integral encoding we trivially set \( f_0(x) = 1 \). Note that we can efficiently decide the sign (and phase) of the trivial function given we need only test the four possibilities for a phase and we can efficiently verify the result of our state-preparation procedure via the destructive interference circuit in Fig. 10 as discussed in Appendix E.3.

**Corollary 2.** By Lemma 3, if we re-scale \( f_1 \) as per Remark 3, the bound on the spectral gap of the Hamiltonian may be written as,

\[
g(s) \geq \frac{1}{2},
\]

which is indeed a constant.

**Lemma 5 (Delay factor bound).** Given the Hamiltonian \( \mathcal{H}(s) \) as defined in Lemma 4, we have the bound

\[
\left\| \frac{d}{ds} \mathcal{H}(s) \right\|_2 \leq 2 \cdot \frac{(N(1)^2/N + 1)^2}{\left( N(1)/\sqrt{N} \right)^3}.
\] (C9)

This implies that the total adiabatic evolution time required for the procedure is asymptotically given by \( T \in O \left( \frac{(N(1)^2/N + 1)^2}{\left( N(1)/\sqrt{N} \right)^3} \right) \).

**Proof.** We begin by providing an upper-bound for \( \left\| \frac{d}{ds} \mathcal{H}(s) \right\|_2 \).

\[
\left\| \frac{d}{ds} \mathcal{H}(s) \right\|_2 = \left\| \left( \frac{d}{ds} |v_s\rangle \right) \langle v_s | + |v_s\rangle \left( \frac{d}{ds} \langle v_s | \right) \right\|_2 = \left\| (|v_1\rangle - |v_0\rangle) \langle v_s | + |v_s\rangle (|v_1\rangle - |v_0\rangle) \right\|_2.
\] (C10)

For ease of notation, allow \( |v'\rangle = |v_1\rangle - |v_0\rangle \). Using the triangle inequality,

\[
|||v'\rangle \langle v_s | + |v_s\rangle \langle v'|\rangle_2 \leq ||v'\rangle \langle v_s ||_2 + ||v_s\rangle \langle v'|\rangle_2 = 2 ||v'\rangle \langle v_s ||_2.
\] (C11)

Suppose we have two normalized vectors \( |a'\rangle = \frac{|a\rangle}{\sqrt{\langle a|a \rangle}} \) and \( |b'\rangle = \frac{|b\rangle}{\sqrt{\langle b|b \rangle}} \). Then,

\[
\left\| |a\rangle \langle b| \right\|_2 = \sqrt{\langle a|a \rangle \sqrt{\langle b|b \rangle}} \left\| a' \langle b'| \right\|_2 \leq |||a\rangle ||_2 |||b\rangle ||_2,
\] (C12)

where the inequality follows from \( |||a\rangle ||_2 \leq 1 \) for any normalized vectors \( |a'\rangle \) and \( |b'\rangle \). Applying this to Equation C11 yields,

\[
\left\| \frac{d}{ds} \mathcal{H}(s) \right\|_2 \leq 2 \left\| |v'| \right\|_2 |||v_s\rangle ||_2.
\] (C13)

Of course,

\[
|||v'| \rangle \right\|_2^2 = \langle v' | v' \rangle = \langle v_0 | v_0 \rangle + \langle v_1 | v_1 \rangle - 2 \text{Re} \langle v_0 | v_1 \rangle \leq |||v_0\rangle ||_2^2 + |||v_1\rangle ||_2^2,
\] (C14)

where the inequality follows from the fact that \( \text{Re} \langle v_0 | v_1 \rangle \geq 0 \) given we have properly chose the phase of the trivial function. Thus,

\[
|||v'| \rangle \right\|_2 \leq \sqrt{\frac{N(0)^2}{N} + \frac{N(1)^2}{N}} = \sqrt{1 + \frac{N(1)^2}{N}}.
\] (C15)
where we again used the fact that its trivial to select an $f_0$ (or $g_0$) such that $N(0) = \sqrt{N}$. Equation C13 then becomes,

$$\left\| \frac{d}{ds} \mathcal{H}(s) \right\|_2 \leq 2 \sqrt{1 + \frac{N(1)^2}{N}} \|v_s\|_2.$$  \hspace{1cm} (C16)

Noting that $g(s) = |||v_s|||^2_2$, we can then write

$$\frac{\|d/du\mathcal{H}(s)\|_2}{g(s)^2} \leq \frac{2\sqrt{1 + \frac{N(1)^2}{N}}}{|||v_s|||^3_2}. $$ \hspace{1cm} (C17)

From Lemma 4 we directly obtain,

$$|||v_s|||^3_2 \geq \frac{N(1)^3}{(N(1)^2 + N)^{3/2}}.$$ 

(C18)

Thus,

$$\frac{\|d/du\mathcal{H}(s)\|_2}{g(s)^2} \leq \frac{2\sqrt{N + N(1)^2}}{\sqrt{N}} \frac{(N(1)^2 + N)^{3/2}}{N(1)^3} = \frac{2(N(1)^2 + N)^2}{\sqrt{NN(1)^3}}.$$ 

(C19)

As discussed in Section B3 a sufficiently slow Hamiltonian evolution schedule is given by

$$\tau(s) \gg \frac{\|d/du\mathcal{H}(s)\|_2}{g(s)^2}.$$ 

(C20)

Thus, $T = \int_0^1 \tau(s) ds$, and so asymptotically we also have the bound $T \in O\left( \frac{(N(1)^2+N)^2}{\sqrt{NN(1)^3}} \right)$.

\[\square\]

**Corollary 3.** By Lemma 7, if we re-scale $f_1$ as per Remark 5, the bound on the delay factor of $\mathcal{H}(s)$ is given by,

$$\|d/du\mathcal{H}(s)\|_2 \leq 8.$$ 

Thus, the total adiabatic evolution time required by the procedure is asymptotically constant, i.e. $T \in O(1)$.

b. Time evolution error bounds

**Lemma 6** (Adiabatic discretization error bound). Given the parameterized Hamiltonian $\mathcal{H}(s)$ from Eq. (C1) and its discretised approximation $\mathcal{H}'(s) = \mathcal{H}(\lfloor sr \rfloor / r)$ with a resolution $r \in \mathbb{N}$, the distance between the approximate and exact Hamiltonians is bounded as (in terms of the spectral norm)

$$\|\mathcal{H}(s) - \mathcal{H}'(s)\|_2 \leq \delta_0 \equiv \frac{2}{r} \left(1 + 3\frac{N(1)}{\sqrt{N}} + 2\frac{N(1)^2}{N}\right).$$

This may be equivalently written as,

$$\|\mathcal{H}(s) - \mathcal{H}'(s)\|_2 \leq \frac{2}{r} \left(1 + 3\|A(1)/N\|_2^{1/2} + 2\|A(1)/N\|_2\right).$$ 

Proof. First, observe that the deviation between the exact and approximate Hamiltonians is maximized when $s = \frac{j+1}{r}$ (for $j \in \mathbb{Z}$ and $1 \leq j \leq r$), as $\mathcal{H}'(\frac{j+1}{r}) = \mathcal{H}(\frac{j}{r}).$ Thus,

$$\|\mathcal{H}'(s) - \mathcal{H}(s)\|_2 \leq \left\| \mathcal{H}(\frac{j}{r}) - \mathcal{H}(\frac{j+1}{r}) \right\|_2.$$ 

(C21)

For notational brevity, allow $X_0 = |v_0\rangle\langle v_0|$, $X_1 = |v_0\rangle\langle v_1| + |v_1\rangle\langle v_0|$, and $X_2 = |v_1\rangle\langle v_1|$. Then,

$$\mathcal{H}(s) = |v_s\rangle\langle v_s| = (1 - s)^2 X_0 + s(1 - s) X_1 + s^2 X_2.$$ 

(C22)
Allowing $s = \frac{i+n}{r}$ and $\gamma = \frac{t}{r}$, simple computer algebra shows
\[
\|\mathcal{H}(\gamma) - \mathcal{H}(s)\|_2 = \left\| \left(1 + \frac{2i - 2r}{r^2}\right) (X_0 - X_1 + X_2) + \frac{1}{r} (X_1 + 2X_2) \right\|_2. \tag{C23}
\]
Then,
\[
\|\mathcal{H}'(s) - \mathcal{H}(s)\|_2 \leq \frac{1}{r^2} \|X_0 - X_1 + X_2\|_2 + \frac{2}{r} \|X_0 - X_1 + X_2\|_2 + \frac{1}{r} (\|X_1\|_2 + 2 \|X_2\|_2) \tag{C24}
\]
\[
\leq \frac{1}{r^2} (\|X_0\|_2 + \|X_1\|_2 + \|X_2\|_2) + \frac{1}{r} (2 \|X_0\|_2 + 3 \|X_1\|_2 + 4 \|X_2\|_2) \tag{C25}
\]
where we used the triangle inequality, and the fact that $\frac{1+2i-2r}{r^2} \leq -(1-2r)/r^2$ (since $r \geq 1$). Noting that $\|X_0\|_2 = 1$, $\|X_1\|_2 \leq 2 \frac{\mathcal{N}(1)}{\sqrt{N}}$, $\|X_2\|_2 = \frac{\mathcal{N}(1)^2}{N}$, we obtain
\[
\|\mathcal{H}'(s) - \mathcal{H}(s)\|_2 \leq \frac{2}{r} \left(1 + 3 \frac{\mathcal{N}(1)}{\sqrt{N}} + 2 \frac{\mathcal{N}(1)^2}{N}\right) - \frac{1}{r^2} \left(1 + 2 \frac{\mathcal{N}(1)}{\sqrt{N}} + \frac{\mathcal{N}(1)^2}{N}\right). \tag{C26}
\]
Finally, using $\|A(1)/N\|_2 \approx \mathcal{N}(1)^2/N$, we obtain
\[
\|\mathcal{H}'(s) - \mathcal{H}(s)\|_2 \leq \frac{2}{r} \left(1 + 3 \|A(1)/N\|_2^{1/2} + 2 \|A(1)/N\|_2\right) - \frac{1}{r^2} \left(1 + 2 \|A(1)/N\|_2^{1/2} + \|A(1)/N\|_2\right). \tag{C27}
\]
Since all of the terms quadratic in $r$ are negative, we can then directly drop them to obtain the simplified expression for the upper-bound presented in the lemma.

**Corollary 4.** By Lemma 6, if we re-scale $f_1$ as per Remark 5 the bound on the maximum deviation of the discretized and exact Hamiltonians is given by,
\[
\|\mathcal{H}(s) - \mathcal{H}'(s)\|_2 \leq \frac{8}{r}. \tag{C28}
\]

So far, we have shown that the adiabatic evolution need only run for a constant period of simulation time $T$ (with the constant factor depending on properties of the function), and that the discretization of the adiabatic Hamiltonian may be performed with arbitrarily small error. We now show that the simulation of each $e^{-i\mathcal{H}(s)T}$ term may also be performed efficiently (and with bounded error) and thus that the overall algorithm is efficient.

**Lemma 7 (Error of low-rank simulation).** We apply the low-rank simulation from Statement 4 for time $\Delta t$. The quantum state after measuring the ancilla in the $|+\rangle$ state and discarding the ancilla is $|\psi\rangle = e^{-i\Delta t A/N} |\psi\rangle + |\mathcal{E}\rangle$ with $\|\mathcal{E}\| \leq 5/2 \Delta t^2 \|A\|_{max}$ and with bounded probability $\text{Prob} \geq 1 - 2\Delta t^2 \|A\|_{max}^2$ expressing only leading terms in $\Delta t$. Given the above expression holds for any input state $|\psi\rangle$ it follows that the algorithmic error $\|U(\Delta t) - U(\Delta t)\| := \epsilon_0$ of the approach is bounded as
\[
\epsilon_0(\Delta t) \leq 5/2 \Delta t^2 \|A\|_{max}
\]
where $U'$ is the ideal, piecewise constant unitary $e^{-i\Delta t A/N}$ and $U''$ represents the mapping of our procedure (1-sparse evolution then Hadamard measurement).

**Proof.** Using the notations from Appendix E we explicitly compute the joint state of the ancilla and the main register $|\Psi(t)\rangle = e^{-itS_A} |+\rangle |\psi\rangle$ up to order $t^2$ and neglecting possible digitisation error in computing entries of $S_A$ via an oracle (which can be suppressed arbitrarily) as
\[
|\Psi(t)\rangle = |+\rangle |\psi\rangle - itS_A |\Psi(0)\rangle + O(t^2) = |+\rangle |\psi\rangle - it \sum_{jk} [A]_{jk} |\psi\rangle_{jk} |j\rangle |k\rangle + O(t^2)
\]
where we have used the action of $S_A$ on the amplitudes $\Psi_{jk} := (j|k|)|+\rangle |\psi\rangle$ from Eq. (E1). Substitute the explicit form of the matrix $A_{jk} := \phi_k \phi_j^*$ and thus find the composite state as
\[
|\Psi(t)\rangle = |+\rangle |\psi\rangle - it \sum_{jk} \phi_k \phi_j^* \psi_{jk} |j\rangle |k\rangle + O(t^2) = |+\rangle |\psi\rangle - it |\chi\rangle |\phi\rangle + \epsilon, \tag{C28}
\]

where we used that \( \Psi_{jk} = \psi_j / \sqrt{N} \) and \( \Psi_{kj} = \psi_j / \sqrt{N} \), and we have denoted the ancilla state \( |\chi\rangle := \sum_j \phi_j^* \psi_j / \sqrt{N} |I\rangle \).

In Eq. (C28) we can explicitly write the residual term as \( |\epsilon\rangle := -\frac{\epsilon^2}{T^2} S^2_A |\psi\rangle + \mathcal{O}(t^3) \) and we can bound its vector norm as \( \|\epsilon\| \leq \frac{\epsilon^2}{T^2} \|S_A\|^2 + \mathcal{O}(t^3) \).

Let us now compute the quantum state that we obtain after projecting onto the \(+n\) state on the ancilla qubit with \( \mathcal{P} := (+n\rangle \langle +n| \otimes \text{Id} \) as

\[
\mathcal{P} |\Psi(t)\rangle = |+n\rangle \psi - \frac{it}{N} |+n\rangle \langle +n| A \psi + |\mathcal{P}| \epsilon \rangle,
\]

where we have used that \( \langle +n| \psi \rangle = \sum_k \phi_k^* \psi_k / N \) and note that \( |A \psi\rangle := A |\psi\rangle = \langle \phi | \psi \rangle \).

We compute the probability of this measurement by computing the norm of the projected vector as

\[
\text{Prob} = \left( \langle \Psi(t) | \mathcal{P} \right) \left( \mathcal{P} |\Psi(t)\rangle \right) = \langle +n | \psi \rangle + \frac{it}{N} \langle +n | A \psi \rangle + \langle \epsilon | \mathcal{P} \right) \left( |+n\rangle \psi - \frac{it}{N} |+n\rangle A \psi + |\mathcal{P}| \epsilon \rangle \rangle
\]

\[
= 1 + \frac{t^2}{N^2} (\langle +n | A \psi \rangle |+n\rangle A \psi + 2 \text{Re} \left( \langle +n | \psi \rangle |\mathcal{P}| \epsilon \rangle \right) + \mathcal{O}(t^3).
\]

We will drop the second term in our lower bound given it is non-negative (as it expresses the vector norm \( \|\langle +n | A \psi\rangle\|^2 \)).

We can lower bound the third term by first explicitly expressing it

\[-\langle +n | \psi \rangle |\mathcal{P}| \epsilon \rangle = \frac{\epsilon^2}{T^2} (\langle +n | \psi \rangle |+n\rangle A \psi + \frac{A^2}{N^2} |\psi\rangle \rangle \leq \frac{\epsilon^2}{T^2} \|S_A\|^2 = \frac{\epsilon^2}{T^2} \|A\|_{\text{max}}^2,
\]

where we have used the projector acts trivially on \(+n\rangle \psi\rangle \) and used from ref. [22] that the absolute largest eigenvalue of \( S_A \) is \( \|A\|_{\text{max}} \). We finally conclude that the probability is lower bounded as

\[
\text{Prob} \geq 1 - t^2 \|A\|^2_{\text{max}} + \mathcal{O}(t^3).
\]

(C29)

We can therefore discard the ancilla register (given it is separable and in the \(+n\) state) and we obtain the quantum state \(|\psi'\rangle\) as

\[
|\psi'\rangle = |+n\rangle e^{-itA/N} |\psi\rangle + |\mathcal{P}| \epsilon \rangle + \frac{\mathcal{P}|\epsilon\rangle + \frac{\epsilon^2}{T^2} |+n\rangle \langle A^2/N^2 |\psi\rangle \rangle + \mathcal{O}(t^3),
\]

(C30)

The vector norm of the error term is upper bounded via the triangle inequality as

\[
\| \mathcal{P}|\epsilon\rangle + \frac{\epsilon^2}{T^2} |+n\rangle \langle A^2/N^2 |\psi\rangle \rangle \| \leq \|\epsilon\| + \frac{\epsilon^2}{T^2} \|A\|^2/N^2 \leq 3/2t^2 \|A\|^2_{\text{max}} + \mathcal{O}(t^4)
\]

where we have used the inequality \( \|A\|^2/N^2 \leq \|A\|^2_{\text{max}} \) the Taylor expansion of the probability \( 1/\sqrt{\text{Prob}} = 1/\sqrt{1-x} = 1 + x + \mathcal{O}(x^2) \) with \( x = \mathcal{O}(t^2) \) and the norm of the residual term \( \|\epsilon\| \leq \frac{\epsilon^2}{T^2} \|A\|^2_{\text{max}} + \mathcal{O}(t^3) \).

Furthermore, note that

\[
|\psi'\rangle = e^{-itA/N} |\psi\rangle + |\mathcal{E}\rangle
\]

with \( \|\mathcal{E}\| \leq t^2 \|A\|^2_{\text{max}} + \mathcal{O}(t^3) \) via the previous expansion of the square root function. As such, after measuring the ancilla in the \(+n\) state we obtain the state

\[
|\psi'\rangle = e^{-itA/N} |\psi\rangle + |\mathcal{E}\rangle
\]

with \( \|\mathcal{E}\| \leq 5/2t^2 \|A\|^2_{\text{max}} \) where we have again used the Taylor expansion of the square root function.

\[
\square
\]

**Lemma 8 (Total success probability).** We use the low-rank approach to simulate an evolution for overall time \( T \) by applying \( r \) piece-wise constant evolutions with time \( T/r \). The probability that throughout \( r \) consecutive iterations we always measure the all plus state is lower bounded by

\[
\text{prob}_{\text{tot}} \geq 1 - T^2 \|A\|^2_{\text{max}}/r + \mathcal{O}(T^4/r^2),
\]

and we discard all other measurement outcomes for the sake of analytical simplicity.
Proof. We can compute the overall probability for \( r \) iterations using Eq. (C29) and substituting that \( t = \Delta t = T/r \) as
\[
\text{Prob}' \geq (1 - T^2 \|A\|^2_{\text{max}}/r^2)^r + O(T^3/r^3) = 1 - T^2 \|A\|^2_{\text{max}}/r + O(T^4/r^2).
\] (C31)

Note that the above is a loose bound – in practice one does not need to discard the non-+outcomes as they lead to nearly the same performance; but in Lemma 9 we only took into account and bounded the \(|+\rangle\) outcome fidelities, and bounding other measurement outcomes is beyond the scope of the present work. We note that the probabilistic aspect of the algorithm is simply an analytical convenience, rather than an intrinsic property of the approach.

Lemma 9 (Piecewise simulation error). For a fixed \( s \), the algorithmic simulation error from using Hamiltonian simulation technique summarized in Statement 1 to simulate \( \mathcal{H}(s) \) for a time \( T/r \) is bounded by,
\[
\epsilon_0(s) \leq O \left( \frac{1}{r^2} \left( \frac{\|A(1)\|_{\text{max}}}{\|A(1)/N\|_2} \frac{(1 + \|A(1)/N\|_2)^3}{\|A(1)/N\|^{3/2}_2} \right)^{2} \right).
\]

Proof. As summarised in Statement 1, the low-rank simulation approach of ref [22] for time \( \Delta t \) under \( S_A \) generates the following dynamics under the partial trace from Eq. (B19) as
\[
\sigma' = e^{-i \Delta t A(s)/N} \sigma e^{i \Delta t A(s)/N} + O(\Delta t^2)
\]
for any pure initial state \( \sigma = |\phi\rangle \langle \phi| \). In Lemma 8 we assumed that measurements are performed on the ancilla register and only the \(|+\rangle\) outcome is accepted after which we discard the ancilla register. This leads to an error in terms of the deviation between the exact \( U'(\Delta t, s) := e^{-i \Delta t A(s)/N} \) and obtained unitary mapping \( \| U'(\Delta t, s) - U''(\Delta t, s) \|_2 = \epsilon_0(\Delta t, s) \). For ease of notation we denote the error \( \epsilon_0(s) := \epsilon_0(\Delta t, s) \) as
\[
\epsilon_0(s) \leq \frac{s^2}{2} \|A(s)\|^2_{\text{max}} \Delta t^2.
\] (C32)

In the direct-state preparation algorithm, we must evolve for a total time \( T \), which is constant bounded, as shown in Lemma 5. We do so in \( r \) evenly spaced intervals, resulting in \( \Delta t = T/r \). As a result,
\[
\epsilon_0(s) \leq \frac{s^2}{2} \left( \frac{\|A(s)\|_{\text{max}}}{\|A(1)/N\|_2} \right)^2 \left( \frac{T}{r} \right)^2.
\] (C33)

Noting that \( f_s \) has a maximum filling ratio when \( s = 0 \), for large \( N \) we have,
\[
\frac{\|A(s)\|_{\text{max}}}{\|A(1)/N\|_2} \leq \frac{\|A(1)\|_{\text{max}}}{\|A(1)/N\|_2}.
\] (C34)

Observe that
\[
\|A(s)/N\|_2 \leq (1 - s)^2 \langle v_0 | v_0 \rangle + s^2 \langle v_1 | v_1 \rangle + 2s(1 - s) \text{Re} \langle v_0 | v_1 \rangle
\] (C35)
\[
\leq (1 - s)^2 + s^2 \frac{N(1)^2}{N} + 2s(1 - s) \frac{N(1)}{\sqrt{N}}
\] (C36)
\[
= \left( 1 - s + s \frac{N(1)}{\sqrt{N}} \right)^2
\] (C37)
\[
\leq \max \left( 1, \frac{N(1)^2}{N} \right)
\] (C38)
\[
\leq 1 + \frac{N(1)^2}{N},
\] (C39)

where the second inequality follows from the application of the Cauchy-Schwarz inequality on \( \langle v_0 | v_1 \rangle \). Combining Equations C33 and C34, we obtain the bound
\[
\epsilon_0(s) \leq \frac{s^2}{2} \left( \frac{\|A(1)\|_{\text{max}}}{\|A(1)/N\|_2} \right)^2 \left( \frac{\|A(s)/N\|_2}{\|A(1)/N\|_2} \right)^2 \left( \frac{T}{r} \right)^2
\] (C40)
\[
\leq \frac{s^2}{2} \left( \frac{\|A(1)\|_{\text{max}}}{\|A(1)/N\|_2} \right)^2 \left( 1 + \frac{N(1)^2}{N} \right) \left( \frac{T}{r} \right)^2
\] (C41)
where the inequality follows from using Equation \([C39]\). Using Lemma \([3]\), we have an upper-bound on the delay factor, and thus an asymptotic upper-bound on \(T\). To allow treatment as a strict upper-bound, we introduce the constant \(k\), such that \(T \leq 2k(\frac{N(1)^2 + N}{\sqrt{NN(1)^3}})\) (in practice, we find that \(k \approx 100\)). Of course, \(\left(1 + \frac{N(1)^2}{N}\right)^3 = \frac{1}{N} (N + N(1)^2)\). Then,

\[
\epsilon_0(s) \leq \frac{10k^2}{r^2} \left( \frac{\|A(1)/N\|_2}{\|A(1)/N\|_2} \right)^2 \left( \frac{\|A(1)/N\|_2}{\|A(1)/N\|_2} \right)^3 \left( \frac{\|A(1)/N\|_2}{\|A(1)/N\|_2} \right)^3
\]

\[
= \frac{10k^2}{r^2} \left( \frac{\|A(1)/N\|_2}{\|A(1)/N\|_2} \right)^2 \left( \frac{\|A(1)/N\|_2}{\|A(1)/N\|_2} \right)^3 \left( \frac{\|A(1)/N\|_2}{\|A(1)/N\|_2} \right)^3
\]

\[
= \frac{10k^2}{r^2} \left( \frac{\|A(1)/N\|_2}{\|A(1)/N\|_2} \right)^2 \left( \frac{\|A(1)/N\|_2}{\|A(1)/N\|_2} \right)^3 \left( \frac{\|A(1)/N\|_2}{\|A(1)/N\|_2} \right)^3
\]

where we used the fact that \(\|A(1)/N\|_2 = \frac{N(1)^2}{N}\).

**Lemma 10.** The query complexity of simulating \(e^{-i\mathcal{H}(s)\Delta t}\) as per Lemmas \([2]\) and \([4]\) is simply 4, i.e. is bounded by \(O(1)\), while the total number of qubits required are \(2n + d + 2\), and thus the algorithm requires \(O(n + d)\) ancillary qubits. Therefore, the total state preparation algorithm has query complexity \(O(r)\).

**Proof.** As discussed at length, the simulation of \(e^{-i\mathcal{H}(s)\Delta t}\) is implemented using the low-rank Hamiltonian simulation technique of ref \([22]\), where they embed the \(N \times N\) matrix \(A_s \equiv \mathcal{H}(s)\) into the one-sparse \(N^2 \times N^2\) matrix \(S_{A_s}\). The one-sparse matrix can then be simulated as described in Lemma \([1]\) using a total of 4 calls to the oracles \(O_f\) and \(O_H\), and with a total of \(2n + d + 2\) qubits (i.e. \(n + d + 2\) ancillary qubits). As the algorithm consists of performing \(r\) consecutive such Hamiltonian simulations, the total query complexity is given by \(O(r)\).
3. Asymptotic limits

Above we have derived our bounds in terms of two quantities as the matrix norms $\|A(1)/N\|_2 = N(1)^2/N$ and $\|A(1)\|_{max,2}$. We prove in Appendix C1 that both these quantities naturally admit asymptotic limits for $N \to \infty$ for any continuous function. Furthermore, we also prove that Lipschitz continuous functions—which cover nearly all instances in practice—approach these limits in exponential order in the number of qubits $n$ as illustrated in Fig. 7 and in Fig. 8. We apply these results and establish asymptotic limits of our error bounds.

Remark 6 (Growth rate of normalisation). For any fixed continuous function $f$, we obtain the scaling for our pointwise encoding $N(1) \in O(\sqrt{N})$ in the asymptotic limit of big $N$. More generally we can state $N(1)/\sqrt{N} \in \Theta(1)$, which immediately follows from the fact that the normalisation constant asymptotically approaches the Riemann integral $N(1)^2/N \propto \|f_1\|_2^2$ as per Lemma 13, Lemma 14 and Lemma 15.

These asymptotic limits are illustrated in Fig. 7. Intuitively, each $\epsilon$ neighbourhood around any $f_1(x)$ is asymptotically constant and so the summation for $N(1)$ is a sum of $O(N)$ constants and therefore taking the square root results in a quantity which clearly grows as $O(\sqrt{N})$. As a consequence of this asymptotic limit, we are guaranteed that the total adiabatic evolution time from Lemma 5 is $T \in \Theta(1)$, which property we exploit in the following.

Corollary 5 (Asymptotic simulation error). By Lemma 9, if we re-scale $f_1$ as per Remark 3, the algorithmic simulation error from simulating $H(s)$ for time $T/r$ using the low-rank Hamiltonian simulation technique as per Statement 7...
is bounded by $\epsilon_0(s) \in O(\|A(1)\|_{\text{max}}^2/r^2)$. In the limit of big $N$, this may equivalently be written in terms of the filling ratio $F$ from Definition 3 as

$$
\epsilon_0(s) \in O\left(\frac{F p}{r^2}\right)
$$

where $p = -4$ in the point-wise sampling case, and $p = -2$ in the integration case via Lemma 14 and Lemma 15.

Using the low-rank simulation approach has two significant advantages. First, as the matrix entries of $A$ are computed using arithmetic operations, it is beneficial that the entries have asymptotically constant magnitudes (i.e. $|A|_{kl} \in O(N^0)$). Thus, we can use a fixed resolution in encoding the matrix entries $A_{kl}$ as integers, whereas the desired quantum state might have exponentially small amplitudes. Second, the fact that our simulation actually simulates $A/N$ instead of just $A$ is actually advantageous, as the simulation of $A/N$ automatically absorbs the exponentially decreasing (in terms of the number of qubits) absolute values of the amplitudes of the quantum states (that encode continuous function) — which we prove in Property 3 and in Property 3. In other approaches, encoding the wave function may necessitate using exponentially decreasing rotation angles, but here this is entirely avoided.

**Lemma 12 (Total success probability).** The success probability in Lemma 8 admits the asymptotic limit

$$
\text{prob}_{\text{tot}} \geq 1 - O(F_p/r),
$$

where $p = -4$ in the point-wise sampling case, and $p = -2$ in the integration case via Lemma 14 and Lemma 15.

**Proof.** We obtain the asymptotic bounds by substituting limits from Lemma 14 and Lemma 15 into our bounds in Lemma 8.

**Corollary 6 (Total simulation error).** By Lemma 11 and Remark 6, the total error of the direct state preparation procedure, as measured by the deviation of the exact unitary evolution $U(T)$ from the worst-case unitary evolution we implement $U''(T)$, is given by,

$$
\|U(T) - U''(T)\|_2 \in O\left(\sqrt{\frac{F_p}{r}}\right),
$$

in the limit of big $N$ via Lemma 14 and Lemma 15.

**Proof.** Taking the limit of the bound in Lemma 11 for big $N$, assuming a normalized $f_1$ (although this assumption is not necessary), $T \in O(1)$, $\delta_0 = \frac{1}{r}$, $\epsilon_0(s) \in O(\frac{F_p}{r^2})$, we get the asymptotic upper bound

$$
\|U(T) - U''(T)\|_2 \in O\left(\sqrt{\frac{1 + F_p}{r}}\right).
$$

Noting that $F_p \geq 1$, $1 + F_p \leq 2F_p$, and so our asymptotic statement simplifies to,

$$
\|U(T) - U''(T)\|_2 \in O\left(\sqrt{\frac{F_p}{r}}\right),
$$

(C53)

It is worth explicitly noting that this asymptotic bound also holds when $f_1$ is not normalized. If $f_1$ is not normalized, the ratio $N(1)^2/N$ is still asymptotically $O(1)$, however it may be a large constant, depending on how “unnormalized” $f_1$ is. As a result, this will be off by at most a constant factor, and so the asymptotic bound remains unchanged.

**Remark 7 (When adiabatic error is negligible).** In practical applications (especially for small filling ratios $F$) the total error is not dominated by the error of the discretised adiabatic evolution but rather by the low-rank simulation from Lemma 3 for which we incur an error at each small timestep $\Delta t$ asymptotically via Corollary 3 as $O(F_p/r^2)$. Thus neglecting the adiabatic simulation error and applying the triangle inequality $r$ times consecutively (given $c_0(s)$ is a bound on the unitary operator distance) we obtain the bound

$$
\|U(T) - U''(T)\|_2 \lesssim O\left(\frac{F_p}{r}\right),
$$

where $p = -4$ in the point-wise sampling case, and $p = -2$ in the integration case. This scaling is tighter in $r$ but looser in the filling ratio $F$ than in Corollary 4.
4. Generalisation to arbitrary functions (mappings)

So far we have considered continuous functions $f(x)$ and discretised them to obtain our desired normalised quantum state from Eq. (A3). We proved in Appendix C2 that for any finite qubit count our approach allows us to prepare our final desired state with complexity that depends only on the two matrix norms $\|A(1)/N\|_2 = N(1)^2/N$ and $\|A(1)\|_{max}$, in fact on the ratio of these two. We did not actually have to invoke the property that the function is continuous but far. In fact we invoked continuity because it guarantees that asymptotically in $N$ the matrix norm $\|A(1)\|_{max} \rightarrow \max_x |f(x)|^2$ is approached via the boundedness theorem and that $\|A(1)/N\|_2 \rightarrow (b-a)\|f\|^2_2$ due to Riemann integrability as we prove in Lemma 14. For this reason we do not actually need to necessary assume continuity of the function as we explain now.

In fact our proofs in Appendix C2 are applicable to the case of any discrete mapping as $F^{(n)} : \{0,1\}^n \mapsto \mathbb{C}$ in the sense that we can efficiently prepare any $n$-qubit quantum state of the form

$$\ket{\psi}_n = (\mathcal{N}_N)^{-1} \sum_{j \in \{0,1\}^n} F^{(n)}(j)\ket{j}_n,$$

(C54)

given the following conditions are satisfied.

**Theorem 3.** Given an arbitrary efficiently computable mapping $F^{(n)} : \{0,1\}^n \mapsto \mathbb{C}$ that depends on the qubit count $n$ with absolute maximum value $\|F^{(n)}\|_{max} = \max_{k \in \{0,1\}^n} |F^{(n)}(k)|$ and vector norm $\mathcal{N}^2 = \sum_{k \in \{0,1\}^n} |F^{(n)}(k)|^2$ we can prepare the quantum state in Eq. (C54) efficiently using our adiabatic approach given that $\|F^{(n)}\|_{max} \in O(1)$ is asymptotically upper bounded that $\mathcal{N}/\sqrt{N} \in \Theta(1)$ is asymptotically both upper and lower bounded. In direct analogy with Theorem 2 the query complexity then depends on the generalised filling ratio $r \in O(\mathcal{F}^{-4}/\epsilon^2)$.

**Proof.** We define the pointwise encoded matrix $[A(s)]_{kl} := F_s(k)F^*_s(l)$ following Eq. (C1) with the parametric mapping $F_s := (1-s)F_0 + sF_1$ where $F_0$ is the same trivial function with a properly chosen phase as in Appendix C2 and $F_1$ is our desired mapping $F_1 := F^{(n)}$ where we will drop the dependence on $n$ for ease of notation. We proved in Lemma 5 that the total adiabatic evolution time is bounded as $T \in O\left(\frac{(\mathcal{N}(1)^2/N+1)^2}{(\mathcal{N}(1)/\sqrt{N})^2}\right)$ and in Lemma 6 that the simulation error is bounded as

$$\epsilon_0(s) \in O\left(\frac{1}{r^2} \left(\frac{\|A(1)/N\|_{max} (1+\|A(1)/N\|^2_{3/2})^3}{\|A(1)/N\|_2^3}\right)^2\right).$$

for any finite qubit count. As such, given the matrix norms $\|A(1)/N\|_2 = N^2(1)/N$ and $\|A(1)\|_{max} = \|F_1\|_{max}$ are asymptotically in $n$ both lower and upper bounded as $\mathcal{N}(1)/\sqrt{N} \in \Theta(1)$ and upper bounded $\|F_1\|_{max} \in O(1)$ we can guarantee a bounded total evolution time $T \in O(1)$ and that the simulation error is bounded as $\epsilon_0 \in O(1/r^2)$.

Indeed in the special case when $F(j) = f(x_j)$, $F(j)$ is just a discretisation of a continuous function, and the norms are convergent for $n \rightarrow \infty$ as $\|A(1)/N\|_{max} \rightarrow \max_x |f(x)|^2$ is approached via the boundedness theorem and that $\mathcal{N}(1)/\sqrt{N} \rightarrow \sqrt{(b-a)}\|f\|^2_2$ due to Riemann integrability as we prove in Lemma 14. In contrast, our more general asymptotic conditions do not require convergence and it suffices that there exist constants such that the $n$-dependent norms $\mathcal{N}(1)/\sqrt{N}$ and $\|F_1\|_{max}$ are asymptotically bounded. Clearly, in analogy with the filling ratio in Theorem 2 in the more general case the query complexity depends on the generalised filling ratio $r \in O(\mathcal{F}^{-4}/\epsilon^2)$ which we can define as an asymptotic bound on the ratio of norms as

$$\frac{\mathcal{N}(1)}{\sqrt{N}\|F_1\|_{max}} \in O(\mathcal{F}).$$

(C55)

Let us illustrate the above result on two examples.

**Example:** Let us consider a mapping $F^{(n)} : \{0,1\}^n \mapsto [-1,1]$ that takes a binary integer $j \in \{0,1\}^n$ and uses it as a seed to efficiently compute a sample $F^{(n)}(j) = x_j$ from a random distribution $X$ such that the mean is $\mu(X) = 0$. By definition this mapping is bounded $\|F^{(n)}\|_{max} \leq 1$ and we can see that the norm converges to the asymptotic constant

$$\lim_{n \rightarrow \infty} \frac{\mathcal{N}^2}{N} = \lim_{n \rightarrow \infty} \frac{1}{N} \sum_{k \in \{0,1\}^n} x_k^2 = \text{Var}[X].$$
As such, we can efficiently prepare an exponentially large list of $N$ random numbers of zero mean with complexity determined by the variance via the generalised filling ratio $\tilde{F} = \text{Var}[X]$.

**Counterexample:** Let us consider the unstructured search problem, where we have a function $F^{(n)} : \{0,1\}^n \to \{0,1\}$ such that there is only one marked input $m$ for which $F^{(n)}(m) = 1$ and for all other inputs $F^{(n)}(j) = 0$ with $j \neq m$. By definition this mapping is bounded $\|F^{(n)}\|_{\max} = 1$ and we can straightforwardly evaluate the normalisation as

$$\mathcal{N}^2/N = \frac{1}{N} \sum_{k \in \{0,1\}^n} |F^{(n)}(k)|^2 = \frac{1}{N}.$$  

Indeed we find the generalised filling ratio $\tilde{F} = 2^{-n/2}$ and thus the query complexity of preparing this mapping grows exponentially as $O(2^{2n})$. Given efficiently preparing this state would solve the unstructured problem we see it is in fact less efficient than a classical direct search.

**Appendix D: State Preparation via Quantum Phase Estimation**

In this section, we introduce a non-deterministic variant of the state preparation procedure utilizing quantum phase estimation. Begin by defining the two $n$-qubit states,

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{2^n-1} f(x_j) |j\rangle, \quad \text{and} \quad |\phi\rangle = \sum_{j=0}^{2^n-1} f(x_j) |j\rangle$$  \hspace{1cm} (D1)

where $|\psi\rangle$ is normalized (with normalizing coefficient $\frac{1}{\sqrt{N}}$) and $|\phi\rangle$ is not. Our low-rank Hamiltonian simulation procedure is capable of performing the evolution $e^{-i t \frac{F}{N} |\psi\rangle\langle\psi|}$ for some $t$ with error scaling as $O(t^2)$, as per Statement 1. Thus, we define the unitary $U(t) = e^{-i \frac{t}{N} |\psi\rangle\langle\psi|}$, and note that $U(t)^k = U(tk)$.

Noting that $|\phi\rangle\langle\phi| = \mathcal{N}^2 |\psi\rangle\langle\psi|$, we can rewrite $e^{-i \frac{t}{N} |\psi\rangle\langle\psi|} = e^{-i \frac{t}{N} \mathcal{N}^2} |\psi\rangle\langle\psi|$. Suppose $\{|\psi_j\rangle\}_{j}$ forms an orthonormal eigenbasis for $e^{-i \frac{t}{N} \mathcal{N}^2} |\psi\rangle\langle\psi|$, such that $|\psi_0\rangle \equiv |\psi\rangle$ and,

$$e^{-i \frac{t \mathcal{N}^2}{N}} |\psi\rangle\langle\psi| |\psi_j\rangle = e^{-i \gamma_j} |\psi_j\rangle,$$  \hspace{1cm} (D2)

where $\gamma_0 = \frac{t \mathcal{N}^2}{N}$ and $\gamma_j = 0$ for $j > 0$. Then, we create the initial state $|\gamma\rangle$, and expand it in the aforementioned eigenbasis,

$$|\gamma\rangle = \sum_{j=0}^{2^n-1} c_j |\psi_j\rangle,$$  \hspace{1cm} (D3)

where $c_j$ is the complex amplitude of each eigenvector in this basis associated with our state. Moreover, let $t < 0$, so that $-t$ is positive.

Now suppose we have an ancillary $m$-qubit register initially in the state $|+\rangle^\otimes m$. Then the initial joint state of our system is given by,

$$|+\rangle^\otimes m |\gamma\rangle = \sum_{j=0}^{2^n-1} c_j |+\rangle^\otimes m |\psi_j\rangle.$$  \hspace{1cm} (D4)

We now apply a sequence of $m$ controlled $U(t)$ gates, such that when the control is applied on the $j$th qubit in the $m$-qubit register, we apply $U(t)^{2^{m-j-1}} = U(t^{2^{m-j-1}})$ on the $n$-qubit register, as shown in Figure D. Assuming the absence of algorithmic error in the simulation of $U(t)$, this yields the state,

$$\frac{1}{\sqrt{2^m}} \sum_{j=0}^{2^n-1} c_j (|0\rangle + e^{2\pi i \gamma_j 2^{m-1}} |0\rangle) \otimes \ldots \otimes (|0\rangle + e^{2\pi i \gamma_j 2^{0}} |0\rangle) |\psi_j\rangle.$$  \hspace{1cm} (D5)

For ease of explanation, assuming that $\gamma_j$ can be exactly represented in $m$ bits, applying an inverse-QFT in the first register yields,

$$\sum_{j=0}^{2^m-1} c_j |\gamma_j\rangle |\psi_j\rangle = c_0 |\gamma_0\rangle |\psi_0\rangle + |0\rangle \sum_{j=1}^{2^m-1} c_j |\psi_j\rangle,$$  \hspace{1cm} (D6)
where the equality follows from the fact that $\gamma_j \neq 0$ if and only if $j = 0$. Thus, measuring the first register collapses the system to the desired state $|\psi_0\rangle$ with probability $|c_0|^2$ (if $t$ is selected to be sufficiently large – defined shortly) and collapses the state to a superposition of all orthogonal eigenvectors with probability $1 - |c_0|^2$.

We must now consider how a sufficiently large $t$ is selected. In this analysis, we assume the value of $N$ is known (noting that if it is not known, the procedure in the following subsection can be used to find it). If $t$ is selected to be too small, each evolution performed is too close to identity, and the value in the first register after the inverse QFT will always be $|0\rangle$ (and thus our target register does not collapse to the desired state). However, we want to pick the minimum $t$ possible, as in practice, the error in each simulation scales with the square of the total simulation time. We note that the value of the most-significant qubit in the $m$-qubit ancillary register must perform a rotation of at least $2\pi$ for the inverse QFT to yield a non-zero result. As such, since $\gamma_0 = \frac{tN^2}{2}$ and the longest time evolution $U(t^{2m-1})$ we implement should be at least a full period, the minimum possible value of $t$ must satisfy

$$\gamma_0^{2m-1} = 1 \implies \frac{tN^22^{m-1}}{N} = 1. \quad (D7)$$

Thus, we have a bound for the smallest possible $t$ that gives a non-zero result in the first register with probability $|c_0|^2$:

$$t \geq \frac{N}{N^22^{m-1}}. \quad (D8)$$

Moreover, we also note that in QPE, the phase we learn must be between 0 and 1. As such, we have the constraint on the maximum value of $t$ of

$$e^{2\pi i\gamma_0} = e^{2\pi i\frac{tN^2}{2}} \implies t \leq \frac{N}{N^2}. \quad (D9)$$

In order for this procedure to allow the efficient preparation of $|\psi\rangle$, all that remains is to show that it is possible to prepare some initial $\gamma$ such that $|c_0|^2$ is reasonably bounded (and thus that we have an efficient probability of success). To do this, we simply use Property 4 and prepare the state described in Eq. (E1) in Appendix E. Doing so guarantees that $c_0 \geq \mathcal{F}/(b - a)$ and so our probability of successfully preparing the state is lower-bounded by $\mathcal{F}^2/(b - a)^2$ (i.e. it only depends on properties of the function and on the width of the interval being considered). We note that the probability of success presented here is a very loose lower-bound, and there are a number of straightforward ways to boost this.

1. Application: Estimating Normalization Factor

Again, suppose $\{|\psi_j\rangle\}_j$ forms an orthonormal eigenbasis for $e^{-i\frac{tN^2}{2}|\psi\rangle\langle\psi|}$, such that $|\psi_0\rangle \equiv |\psi\rangle$ and, $e^{-i\frac{tN^2}{2}|\psi\rangle\langle\psi|} = e^{-i\gamma_0}|\psi\rangle\langle\psi|$, where $\gamma_0 = \frac{tN^2}{2}$ and $\gamma_j = 0$ for $j > 0$.

Moreover, suppose we have prepared the state $|\tilde{\psi}\rangle$ such that

$$|\tilde{\psi}\rangle = \sqrt{\lambda}|\psi_0\rangle + \sqrt{1 - \lambda}|\psi_\perp\rangle, \quad (D10)$$
where $|\psi_\perp\rangle = \sum_{j=1}^{N-1} c_j |\psi_j\rangle$ and $|\psi_\perp\rangle$ is normalized. We can then apply the phase estimation circuit shown in Figure D assuming perfect error-free simulation, to obtain the state

$$\sqrt{\lambda} |\gamma_0\rangle |\psi_0\rangle + \sqrt{1 - \lambda} \sum_{j=1}^{N-1} c_j |\gamma_j\rangle |\psi_j\rangle.$$  \hfill (D11)

Of course, for $j > 0$, $\gamma_j = 0$, and so this can be simplified to

$$\sqrt{\lambda} |\gamma_0\rangle |\psi_0\rangle + \sqrt{1 - \lambda} |0\rangle |\psi_\perp\rangle.$$  \hfill (D12)

If we have selected a sufficiently large value for $t$ (sufficiently large is defined in Section D), then the first register only collapses to the $|0\rangle$ state when the second register collapses to the $|\psi_\perp\rangle$ state. In this scenario, we measure the first register until we see a non-zero value, which is guaranteed to be $|\gamma_0\rangle$. This value can then be directly used to compute the normalization factor with,

$$N = \sqrt{\frac{N \gamma_0}{t}}.$$  \hfill (D13)

Here, $N$ is clearly known based off of the system size, $t$ is a user specified parameter, and $\gamma_0 \in [0, 1]$ is read out from the quantum register. In the absence of simulation error, naive direct sampling of the quantum state takes an expected number of trials of order $\Theta(\frac{1}{\lambda})$ to measure the state $|\gamma_0\rangle$ and thus to compute the normalization factor. Noting that we can easily create a state $|\tilde{\psi}\rangle$ such that $\lambda \geq \mathcal{F}^2/(b - a)^2$, the expected number of trials required for success is then bounded by $\Omega \left( (b - a)^2 / \mathcal{F}^2 \right)$ (i.e. it depends only on the width of the interval, and on properties of the function).

However, if $t$ is not sufficiently large, then the first register will be unentangled and in the $|0\rangle$ state, regardless of the state of the second register. As a result, no matter how many times you sample the resulting quantum state, you will always measure the first register in the $|0\rangle$ state, and thus gain no information about the normalization factor. Then, if we take $s$ samples from the state in Equation D12 and never observe a non-zero state, we can conclude with high probability that $t$ is too small, so long as $s$ is selected to be a large enough value. Precisely, the probability of failing in any given sample (assuming that $t$ is sufficiently large) is $1 - \lambda$. As a result, the probability of failing after $s$ samples is simply,

$$(1 - \lambda)^s \leq \left( \frac{(b - a + \mathcal{F})(b - a - \mathcal{F})}{(b - a)^2} \right)^s.$$  \hfill (D14)

This motivates an algorithm. Suppose we select an acceptable probability of failure $\epsilon \in (0, 1)$, such that we want to ensure if we conclude $t$ is too small, that we are only wrong with probability at most $\epsilon$. Then we have the bound,

$$\left( \frac{(b - a + \mathcal{F})(b - a - \mathcal{F})}{(b - a)^2} \right)^s \leq \epsilon,$$  \hfill (D15)

which implies we need to perform a number of samples with lower-bound,

$$\left| \frac{\log(b - a + \mathcal{F}) + \log(b - a - \mathcal{F}) - 2 \log(b - a)}{\log \epsilon} \right| \leq s.$$  \hfill (D16)

Thus, $s$ depends logarithmically on constants such as the width of the grid, and properties of the function, and has inverse-logarithmic scaling in the desired probability of failure. Then, if we conclude that $t$ is too small (by taking $s$ samples and observing all $|0\rangle$ outcomes) we can exponentially increase $t$ (e.g. by doubling) and repeat this process until we observe a non-zero measurement in the first register – at which point we can directly compute the value of the normalization factor, as already described. Note that the specific rate at which we increase $t$ each time probably requires some more specific consideration, as we need to be careful to ensure that $t$ never exceeds the maximum value prescribed by Equation D9.

However, if we do not wish to rely on knowledge of properties of the function such as its filling ratio $\mathcal{F}$, we can instead combine this procedure with the adiabatic direct state preparation approach. In doing so, we iteratively construct the state $|\tilde{\psi}\rangle$ by computing the normalization factor for each of the $r$ discretizations of $|\psi(s)\rangle \langle \psi(s)|$ (as defined in Equation C1). In each stage, we compute the normalization factor for $\frac{1}{N} |\psi(s)\rangle \langle \psi(s)|$, and use it to perform the appropriate evolution, giving the state $|\psi(s + \frac{1}{r})\rangle$. We then repeat this process until $s = 1$, at which point we have some initial state that is arbitrarily close to the exact desired state, giving this procedure an arbitrarily high
probability of success. Note that the probability of success does not exponentially decay, as once we have computed the normalization factor at time $s$ we can restart the procedure and deterministically prepare the state up to time $s$.

However, this analysis assumes that we can implement each of the controlled evolutions of $e^{-\frac{i}{\hbar}|\psi\rangle\langle\psi|}$ with no error – in practice, we incur error proportional to $t^2$ for each application of $U(t)$. We leave the analysis of this error prone situation to future work.

2. Application: Efficient Integration of Lipschitz Continuous Functions

The procedure for computing the normalization factor of a function immediately leads to an efficient algorithm for integrating that function over an arbitrary domain. In principle, in the absence of simulation error, this algorithm allows for an exponential improvement in integration over quantum sampling based methods (e.g. using the SWAP test) or any classical methods.

Let us first introduce the approach on the example of a non-negative function. Formally, we wish to integrate the Lipschitz continuous non-negative function $h$ on the interval $[a, b]$, $\int_a^b p(x)dx$. Using Lemma 13, we immediately get

$$\int_a^b h(x)dx = N \sum_{j=0}^{N-1} h(x_j) + \epsilon,$$

where $\epsilon \in O(N^{-1})$ for Lipschitz continuous functions. Define the function $f(x) = \sqrt{h(x)}$. Then,

$$\Delta_N \sum_{j=0}^{N-1} h(x_j) + \epsilon = \Delta_N \sum_{j=0}^{N-1} f(x_j)^2 + \epsilon$$

$$= \Delta_N N^2 + O(N^{-1}),$$

where the last equality follows from the fact that the normalization constant for the function $f$ is given by $N = \sqrt{\sum_{j=0}^{N-1} |f(x_j)|^2}$. Clearly, computing the normalization factor for the function $f$ then immediately leads to an approximation for the integral of $g$ in the domain $[a, b]$ with error $O(N^{-1})$ (in the absence of simulation error). The asymptotic complexity of this procedure is then dominated by the complexity of computing $N$ as described in Subsection D.1.

It follows that any real, Lipschitz continuous function can be integrated this way. We split $f$ into two non-negative functions as $f_+ := (f + |f|)/2$ and $f_- := -(f - |f|)/2$, and apply our approach to these two functions separately. We then subtract the results from each other. This similarly applies to complex functions by splitting them into four quadrants in the complex plane and separately integrating four non-negative functions.

Appendix E: State Preparation via Destructive Interference

1. The Procedure

Here present a simplified version of the above phase-estimation protocol that only requires a single ancilla qubit. Assume we know the normalisation constant $c^2 = \|A/N\|_2 = N^2/N$ we can set an evolution time $t = T/c^2$ and thus we can implement the fixed-time evolution under the projector as $e^{-iT/c^2 A/N} = e^{-iT|\psi\rangle\langle\psi|}$.

Recall that we can write any quantum state in terms of the ideal $|\psi\rangle$ and an orthogonal component as $\sqrt{\lambda}|\psi\rangle + \sqrt{1-\lambda}|\psi_\perp\rangle$ up to trivial global phases. Setting $T = \pi$ allows us to invert the sign of the ideal component $|\psi\rangle$ in any quantum state as

$$e^{-i\pi|\psi\rangle\langle\psi|}[\sqrt{\lambda}|\psi\rangle + \sqrt{1-\lambda}|\psi_\perp\rangle] = -\sqrt{\lambda}|\psi\rangle + \sqrt{1-\lambda}|\psi_\perp\rangle,$$

without affecting the orthogonal component $|\psi_\perp\rangle$ in Hilbert space. Here $\lambda$ is the fidelity with respect to the ideal component.

Applying this evolution as part of the circuit in Fig. 10 allows us to prepare the quantum state and we obtain the
The approach is most efficient when we can initialise $|\psi\rangle$ such that we have a good approximation of $|\psi\rangle$, e.g., by first performing an adiabatic evolution or by efficiently preparing an approximation to $|\psi\rangle$. The success probability of the approach is then completely determined by the quality of the initial state via the fidelity $\lambda = \langle \psi | \psi \rangle$.

As such, upon measuring the ancilla qubit in the $|1\rangle$ state we perfectly obtain $|\psi\rangle$ with probability exactly given by the fidelity $\lambda$.

In principle we can also prepare our desired quantum state just by starting from the all plus state without having an initial approximation to $|\psi\rangle$, i.e., a worst-case approximation. Let us illustrate this on the example of real functions whereby we can perform the procedure in two steps. First we initialise in the all plus state and prepare

$$f_+(x) := |f(x) \pm |f(x)|/2,$$

(E1)

which has a fidelity at least $\lambda \geq F^2/(b-a)^2$ in case of the pointwise encoding via Property 4 with the all plus state for a properly chosen sign. We then prepare $f(x)$ in a second stage which has fidelity at least $1/2$ with respect to $\tilde{f}(x)$. The overall probability of success is then larger then $F^2/(b-a)^2/2$. This ensures us that even in the worst case, by choosing the initial state to be the trivial all-plus state we can guarantee a polynomially bounded (in the filling ratio) success probability.

We can straightforwardly bound the error of the simulation approach.

**Statement 2.** Setting the evolution time as $T = \pi/\|A/N\|_2$ to simulate the time evolution under the projector as $e^{-iTA/N} = e^{-i\pi|\psi\rangle\langle\psi|}$ in $r$ consecutive steps, the algorithmic error is generally bounded as $|\epsilon| \leq O(F^p/r)$ where $p = -2, -4$ depending on the encoding and $F$ is the filling ratio from Definition 3.

**Proof.** Given the normalisation $c^2 = \|A/N\|_2 = \Lambda^2/N$ we use the low-rank simulation approach to implement the piecewise evolutions $e^{-i\Delta t/c^2A/N}$ for $\Delta t = T/r$ and repeat the procedure $r$ times to simulate the dynamics under the projector as $e^{-iTA/N} = e^{-i\pi|\psi\rangle\langle\psi|}$. We have already established the simulation error for a small time step $\Delta t = T/r$ in Corollary 5 in terms of a unitary operator distance as

$$\epsilon_0(s) \in O\left(\frac{F^p}{r^2}\right).$$

When repeating the small time simulation $r$ times the error accumulates via a triangle inequality at worst as $|\epsilon| \leq r|\epsilon_0| \in O(F^p/r)$.

### 2. Application: approximating the normalisation factor

If we do not exactly know the normalisation $c^2 = \|A/N\|_2 = \Lambda^2/N$ we can approximate it the following way. Applying the circuit from Fig. 10 but with a parametric time evolution $e^{-i\pi A/N}$ where we vary $t$ we obtain the
following quantum state immediately before measurement

$$|\Psi'\rangle = [1 - e^{-itc^2}]\sqrt{\lambda}|\psi\rangle \otimes |1\rangle/2 + \ldots.$$  \hfill (E2)

The probability of the outcome $|1\rangle$ is

$$\text{prob}_1(t) = \lambda|1 - e^{-itc^2}|^2/2 = \lambda(1 - \cos tc^2).$$  \hfill (E3)

As such, by estimating the probability $\text{prob}_1(t)$ for an increasing $t$ and, e.g., curve fitting or Fourier transforming the time-dependent signal $\text{prob}_1(t)$, allows us to obtain an estimate of $c^2$ which is subject to shot noise and also to the algorithmic error in Statement 2. Note that none of the techniques in the present work need an exact knowledge of $c^2$ and a rough estimate is always sufficient.

In particular, note that even if we do not exactly know $c^2$ we still obtain exactly the desired state in Eq. (E2) after measuring the $|1\rangle$ ancilla outcome which we obtain with a probability $\text{prob}_1(t)$ that is non-zero as long as $t \neq n2\pi/c^2$ for any $n \in \mathbb{N}$. Indeed the maximum of the probability is at $t = \pi/c^2$.

3. Application: efficient state verification

Given a state $|\phi\rangle = \sqrt{\lambda}|\psi\rangle + \sqrt{1-\lambda}|\psi\rangle_\perp$ that we have for example prepared via the adiabatic approach in Appendix B such that $\lambda \approx 1$, we can use circuit in Fig. 10 to efficiently verify that we have indeed prepared the desired state and, e.g., the hyperparameters of adiabatic evolution are properly chosen: we perform the ancilla measurement in Fig. 10 and we indeed obtain the quantum state $|\psi\rangle$ with probability $\text{prob}_1 = \lambda$ (up to the algorithmic error of the time evolution). When the ancilla measurement returns the $|0\rangle$ outcome then we need to restart the procedure. The probability that we still fail after $m$ trials then decreases exponentially as $(1 - \lambda)^m$. Another significant advantage of this technique is that we can this way remove/project out the algorithmic error of adiabatic evolution due to finite, discretised time evolution at the cost of introducing a small probability of failure − non-deterministic approach.

Appendix F: Details of Fig. 3

In this section we discuss how we numerically simulated our state-preparation approach in Fig. 3. Note that the low-rank simulation approach we utilise requires two quantum registers. In our implementation we explicitly represent the two quantum registers of $n$ qubits each which we use to perform the exact 1-sparse simulation of $S_A$ for a small time step $\Delta t$. We thus numerically exactly simulate the low-rank simulation approach and thus explicitly take into account all error sources of our state preparation approach. After each time step we reset the ancilla state by first performing Hadamard gates and then measuring it in the standard basis.

Via the 1-sparse simulation approach we can exactly simulate the dynamics under a matrix $S_A$ whose matrix entries are given by the function values in Eq. (A3). Given these function values are computed via an oracle as a binary integer in Definition 1 they are thus prone to a (practically negligible) digitisation error when compared to the continuous functions $f: [a, b] \to \mathbb{R}$ we consider in Fig. 3 e.g., $ae^{-\beta|a-x_0|}$. We take this digitisation error into account by assuming $d = 32$ bits resolution and simulate its effect by computing the digitised function values $f(x_k) = \lfloor 2^df(x_k)/\|f\|_{\text{max}} \rfloor/2^d\|f\|_{\text{max}}$ where the symbol $\lfloor \cdot \rfloor$ here stands for rounding to the nearest integer.

In our numerical simulations we took explicitly into account the algorithmic error in the low-rank simulation approach by explicitly simulating the dynamics under $S_A$ over the two registers as

$$|\Psi(t)\rangle = e^{-itS_A}|\Psi\rangle = e^{-itS_A}|+\rangle^n|\psi\rangle,$$

where the state of the ancilla register is $|+\rangle^n := |+\rangle^{\otimes n}$ the uniform superposition. We represent the joint state of the two quantum registers via the rank-2 tensor as $\Psi_{kl} := \langle k|\langle l|\rangle|\Psi\rangle$ with basis states indexed as $k, l \in \{0,1\}^n$. The action of the matrix $|\Psi^{(1)}\rangle := S_A|\Psi\rangle$ can be conveniently computed in terms of the “entry-wise” product of the corresponding tensors as

$$\Psi^{(1)}_{kl} := S_A|\Psi\rangle = (A_{kl}\Psi_{kl})^T = [A_{lk}\Psi_{lk}],$$  \hfill (F1)

where the transpose operation interchanges the indexes of the rank-2 tensor. We can iteratively compute higher powers of the matrix acting on the vector as $|\Psi^{(2)}\rangle = S_A|\Psi^{(1)}\rangle$ etc. via the above formula. This allows us to numerically
compute the effect of the matrix exponential on the tensor using the Taylor expansion without explicitly computing the large matrix \( S_A \) as

\[
|\Psi(t)\rangle = e^{-itS_A}|\Psi\rangle = \sum_{k,l \in \{0,1\}^m} \left[ \sum_{j=0}^{m} \frac{(-it)^j}{j!} \Psi^{(j)}(k)|k\rangle|l\rangle \right],
\]

where we compute the tensor in the square brackets up to \( m = 7 \) to minimise numerical error.

In the numerical simulations we computed the joint quantum state \( |\Psi(\Delta t)\rangle \) for small time steps \( \Delta t \), then performed a Hadamard transformation on the ancilla register \( H_{\otimes n} \otimes \text{Id}||\Psi(\Delta t)\rangle \) and then performed a measurement on the ancilla register in the standard basis. We can accept any outcome and reset the ancilla register accordingly to \(|+\rangle\).

However, note that the probability that the ancilla register collapses into \(|+\rangle\) approaches 1 for small target \( \epsilon \) in Theorem 2 and in Fig. 3 we indeed consider the practically relevant scenario when the state preparation procedure result in the outcome \(|+\rangle\).

\[ \text{Appendix G: Generalisation to multivariate functions} \]

Many applications actually require multivariate input states, such as in quantum chemistry \[13\], quantum field theory \[57\], and derivative pricing with multiple underlying assets \[7\]. However, there is only limited work exploring the preparation of such input states. In 2008, Kitaev and Webb presented an algorithm for preparing multivariate normal theory \[57\], and derivative pricing with multiple underlying assets \[7\]. However, there is only limited work exploring the preparation of general \( d \)-dimensional (i.e., \( d \)-variate) functions then clearly follows by using our approach on \( dn \) qubits and implementing the oracle \( O_f |k\rangle|0\rangle = |k\rangle|f(x_k)\rangle \) for a discretised \( x \) value with \( k \in \{0,1\}^n \). The joint quantum state of the \( d \) registers is then naturally of a tensor product form. In typical applications it may already be a significant advantage to be able to prepare low-dimensional functions to high resolution in separate registers such that the joint quantum state is a separable high-dimensional function. An example can be found in preparing 3-dimensional molecular orbitals in real-space quantum chemistry simulations \[13\]. The initial product state is only an approximation and the quantum computer is then used to entangle the registers anyway, e.g., by performing a simulation between interactive particles that are initially in a separable state.
Appendix H: Further technical details

1. Grover-Rudolph State Preparation

The objective of the Grover-Rudolph state preparation procedure [27] is to prepare the \( n \) qubit state,

\[
|\psi\rangle_n = \sum_{j=0}^{2^n-1} g(x_j)|j\rangle_n,
\]

(H1)

where \( g(x_j) \) is the (normalized) density of the given function \( f \) in the \( j^{th} \) grid interval. In essence, given the correct \( k \) qubit state (with \( 1 \leq k < n \)), they construct the \( k + 1 \) qubit state, iterating this process until the desired \( n \) qubit state is obtained. They then define the function,

\[
h^{(k)}(j) = \frac{\int_{x_j^{(k)}}^{x_{j+1}^{(k)}} f(x)dx}{\int_{x_j^{(k)}}^{x_{j+1}^{(k)}} f(x)dx},
\]

(H2)

where \( K = 2^k \) and \( \Delta K \) is the step-size in a \( k \) qubit grid. That is, \( h^{(k)}(j) \) represents the conditional probability of \( x \) being on the left half of grid interval \( j \) on a \( k \)-qubit grid, given that \( x \) lies in the \( j^{th} \) interval. Clearly, if \( f \) can be integrated efficiently, then \( h^{(k)}(j) \) can be computed efficiently and so can \( \theta_j^{(k)} \equiv \arccos(h^{(k)}(j)) \). An ancillary \( d \) qubit register is then added, as well as a least-significant 1 qubit register. The following state is then obtained by evaluating the oracle performing the mapping \( |j\rangle_k |0\rangle_d \mapsto |j\rangle_k |\theta_j^{(k)}\rangle_d \),

\[
\sum_{j=0}^{2^k-1} g(x_j^{(k)})|j\rangle_k |\theta_j^{(k)}\rangle_d |0\rangle_1.
\]

(H3)

It is important to note that this step requires integration to be performed in superposition. As observed in the Grover paper, any log-concave distribution can be efficiently integrated using Monte Carlo integration, and can thus be integrated efficiently on a quantum computer. Define a CRy gate (with a definition analogous to the CRx gate shown in Figure B1) such that it acts on the last two registers and performs the mapping \( |a\rangle_d |0\rangle_1 \mapsto |a\rangle_d (\cos a|0\rangle_1 + \sin a|1\rangle_1) \). Note that such a gate is derived in Section B (in the general setting, in this case \( t = 1 \)). The CRy gate is then applied to the last two registers in Equation H3 yielding the state

\[
\sum_{j=0}^{2^k-1} g(x_j^{(k)})|j\rangle_k |\theta_j^{(k)}\rangle_d \left( \cos \theta_j^{(k)}|0\rangle_1 + \sin \theta_j^{(k)}|1\rangle_1 \right) = \sum_{j=0}^{2^k-1} g(x_j^{(k)})|j\rangle_k |\theta_j^{(k)}\rangle_d \left( h^{(k)}(j)|0\rangle_1 + (1 - h^{(k)}(j))|1\rangle_1 \right).
\]

(H4)

We then uncompute the \( d \) qubit ancillary register (by applying the inverse of the integration circuit – noting that if the integration is classically stochastic, it must be seeded) and discard it, yielding the state

\[
\sum_{j=0}^{2^k-1} g(x_j^{(k)})|j\rangle_k \left( h^{(k)}(j)|0\rangle_1 + (1 - h^{(k)}(j))|1\rangle_1 \right) = \sum_{j=0}^{2^{k+1}-1} g(x_j^{(k+1)})|j\rangle_{k+1} = |\psi\rangle_{k+1}.
\]

(H5)

The procedure is then repeated until the desired \( n \) qubit state is obtained.

2. Asymptotic properties of the encoded quantum states

Recall that in this work we refer to a continuous function \( f : [a, b] \mapsto \mathbb{C} \) on the closed interval \( a \leq x \leq b \) as \( f(x) \), and will now briefly summarize some notion related to continuous functions which we will frequently utilize in the following proofs. First, recall the general definition of a continuous function \( f \in \mathcal{C} \) as \( \lim_{x \to x_0} f(x) = f(x_0) \) and due to the boundedness theorem all such functions are bounded as \( |f(x)| \leq \|f\|_{\max} \). Second, we define the subset of Lipschitz continuous functions as \( f \in \mathcal{C}_L \subseteq \mathcal{C} \), which are slightly more restrictive as they satisfy \( |f(x_1) - f(x_2)| \leq K|x_1 - x_2| \) for a real constant \( K \geq 0 \), ensuring that they asymptotically approximate any value of the function through discretization in order one \( \mathcal{O}(N^{-1}) \). Third, given all continuous functions are Riemann integrable, it will be natural for us to approximate the finite sum \( \sum_k f(x_k)\Delta_N \) in terms of the integral \( \int_a^b f(x)dx \), for which we incur an asymptotically vanishing approximation error \( \lim_{N \to \infty} \epsilon = 0 \) while for Lipschitz continuous functions we can additionally guarantee the scaling \( \epsilon \in \mathcal{O}(N^{-1}) \) exponential in the number of qubits as we prove now.
Lemma 13 (Riemann sums). Given an arbitrary continuous function $f(x) : [a, b] \rightarrow \mathbb{C}$ on the interval $a \leq x \leq b$, the finite Riemann sum approximates the integral as

$$
\int_a^b f(x) \, dx = \sum_{k=0}^{N-1} f(x_k) \Delta_N + \epsilon
$$

up to an asymptotically vanishing error $\lim_{N \to \infty} \epsilon = 0$. For functions of bounded variation the error term additionally satisfies $\epsilon \in \mathcal{O}(N^{-1})$, see refs. [22, 61] which also applies to Lipschitz continuous functions given they are of bounded variation.

Proof. Given all continuous functions are Riemann integrable we can approximate their Riemann integral via the convergent (for $N \to \infty$) sum as

$$
\int_a^b f(x) \, dx = \sum_{k=0}^{N-1} f(x_k) \Delta_N + \epsilon,
$$

where the error term satisfies $\lim_{N \to \infty} \epsilon = 0$ for all Riemann integrable functions. Here we denote the discretisation interval length as $\Delta_N = (b-a)/N$ and the asymptotic scaling of the approximation error can be established $\epsilon \leq \text{const} \Delta_N$ [22, 61] for a broad class of functions, where the constant factor depends on properties of the functions. For example, if $f$ is differentiable then it is proportional to the largest value of the derivative function within the interval $a \leq x \leq b$. Of course, $f$ need not be differentiable for the above error bound to hold and, e.g., ref [61] established the bound $|\epsilon| \leq V(b-a)/N$ using the absolute largest variation $V$ of the function $f$. Recall that functions of bounded variation $V$ contain all Lipschitz continuous functions. \square

Property 1. Given an arbitrary continuous, non-negative function $f$, the quantum state $|\psi\rangle$ in Eq. (A3) obtained via the integral encoding $g(x)$ becomes identical to the pointwise encoding of the corresponding function $\sqrt{f(x)}$ for large $N$.

Proof. Recall that all continuous functions are Riemann integrable and thus via Lemma 13 we can approximate the Riemann integral as

$$
g(x_j) = \left[ \int_{x_j}^{x_{j+\Delta_N}} f(x) \, dx \right]^{1/2} = \left[ \Delta_N f(x_j) + \epsilon \right]^{1/2}, \quad (H6)
$$

and the error generally satisfies $\lim_{N \to \infty} \epsilon = 0$ and for Lipschitz continuous functions we can also guarantee the scaling $\epsilon \in \mathcal{O}(N^{-1})$. The encoded quantum state thus satisfies

$$
|\psi\rangle = \sum_{j \in \{0,1\}^n} g(x_j) |j\rangle = \sqrt{\Delta_N} \sum_{j \in \{0,1\}^n} \sqrt{f(x_j)} |j\rangle + \epsilon, \quad (H7)
$$

where $\sqrt{\Delta_N}$ approaches the normalisation constant $(N^N)^{-1}$ from a pointwise encoding of $\sqrt{f}$ – and thus indeed the two encodings become identical for large $N$. Note that we satisfy our normalisation condition from Definition 2 that $||\psi||^2 = 1 \approx \sum_{j \in \{0,1\}^n} f(x_j) \Delta_N$. \square

Property 2 (pointwise encoded amplitudes). Given an arbitrary continuous function, the pointwise encoded amplitudes satisfy the general bound $N|\psi_k|^2 \leq (b-a) \|\tilde{f}\|_{\text{max}}^2 / \|f\|_2^2 + \epsilon$. The error term asymptotically vanishes $\lim_{N \to \infty} \epsilon = 0$ and for Lipschitz continuous function it additionally satisfies $\epsilon \in \mathcal{O}(N^{-1})$

Proof. Note that at a level of discretisation $N$ the amplitudes are specified as $|\psi_k|^2 = |f(x_k)|^2 / N^2$, where the normalisation $N^2 = \sum_{k=0}^{N-1} |f(x_k)|^2$ ensures that our quantum state has unit norm. Let us now upper bound the amplitudes as

$$
N|\psi_k|^2 = \frac{N|f(x_k)|^2}{N^2} = \frac{(b-a)|f(x_k)|^2}{N \Delta_N}
$$

It is a direct consequence of Lemma 13 that the normalisation condition $N^2 \Delta_N = \|f\|_2^2 + \epsilon$ can be approximated via the Riemann integral of $|f(x)|^2$. Substituting this back we obtain the formula

$$
N|\psi_k|^2 = \frac{(b-a)|f(x_k)|^2}{\|f\|_2^2 + \epsilon} \leq \frac{(b-a) \|\tilde{f}\|_{\text{max}}^2}{\|f\|_2^2 + \epsilon} = \frac{(b-a) \|f\|_{\text{max}}^2}{\|f\|_2^2} + \epsilon.
$$
Above we have first used the definition of the infinity norm \( \|f\|_{\infty} := \max_x |f(x)| \) and then we have expanded the fraction into a series in \( \epsilon = \mathcal{O}(N^{-1}) \).

The above upper bound is actually saturated given \( f \) is Lipschitz continuous as \( \max_k |f(x_k)|^2 = \|f\|_{\infty}^2 + \mathcal{O}(N^{-1}) \). As such, we can establish that the largest amplitude converges in exponential order to the expression that is independent of the resolution \( N \)

\[
N \max_k |\psi_k|^2 = \frac{(b-a)\|f\|_{\infty}^2}{\|f\|_2^2} + \epsilon.
\]

The error term asymptotically vanishes \( \lim_{N \rightarrow \infty} \epsilon = 0 \) and for Lipschitz continuous function it additionally satisfies \( \epsilon \in \mathcal{O}(N^{-1}) \) via Lemma 13.

### Property 3 (integral encoded amplitudes).

Given an arbitrary continuous function, the integral encoded amplitudes satisfy the general bound \( |\psi|^2 \leq (b-a)\|f\|_{\infty}^2 + \epsilon \). The error term asymptotically vanishes \( \lim_{N \rightarrow \infty} \epsilon = 0 \) and for Lipschitz continuous function it additionally satisfies \( \epsilon \in \mathcal{O}(N^{-1}) \).

**Proof.** Let us approximate the probabilities via the Riemann summation where we only use 1 summand as

\[
|\psi|^2 = g^2(x_k) = \int_{x_k}^{x_k+\Delta N} f(x) \, dx = f(x_k)\Delta N + \epsilon
\]

via Lemma 13. As such we can establish the general bound

\[
\max_k g(x_k) = \max_k |\psi|^2 = \|f\|_{\infty}\Delta N + \epsilon,
\]

Therefore we conclude that

\[
N \max_k g(x_k) = N \max_k |\psi|^2 = (b-a)\|f\|_{\infty}^2 + \epsilon.
\]

The error term asymptotically vanishes \( \lim_{N \rightarrow \infty} \epsilon = 0 \) and for Lipschitz continuous function it additionally satisfies \( \epsilon \in \mathcal{O}(N^{-1}) \) via Lemma 13.

### Property 4 (pointwise overlap with uniform states).

For an arbitrary continuous function \( f(x) : [a,b] \mapsto \mathbb{R} \) on the interval \( a \leq x \leq b \) we construct the non-negative function \( \tilde{f}(x) := |f(x) + |f(x)||/2 \) which has overlap at least 1/2 for a properly chosen sign with our function \( f \) and we encode its function values into a quantum state via our pointwise encoding from Eq. (A3). For any such encoding we can establish the asymptotically bounded fidelity with the easy-to-prepare state \( |+\rangle^{\otimes n} \) as \( |\langle \psi||(+)^{\otimes n}\rangle|^2 \geq F^2/(b-a)^2 + \epsilon \) in terms of our filling ratio \( F \). The error term asymptotically vanishes \( \lim_{N \rightarrow \infty} \epsilon = 0 \) and for Lipschitz continuous function it additionally satisfies \( \epsilon \in \mathcal{O}(N^{-1}) \).

**Proof.** Recall that the normalisation constant form Eq. (A3) satisfies

\[
N^{-1}(\mathcal{N}_N)^2 = \sum_k \tilde{f}(x_k)^2 N^{-1} = \|\tilde{f}\|_2^2/(b-a) + \epsilon,
\]

via Lemma 13. Similarly we can obtain the one-norm of our non-negative function \( \tilde{f} \)

\[
N^{-1} \sum_k \tilde{f}(x_k) = N^{-1} \sum_k |\tilde{f}(x_k)| = \|\tilde{f}\|_1/(b-a) + \epsilon.
\]

The error term asymptotically vanishes \( \lim_{N \rightarrow \infty} \epsilon = 0 \) and for Lipschitz continuous function it additionally satisfies \( \epsilon \in \mathcal{O}(N^{-1}) \) via Lemma 13.

Let us now expand the fidelity \( fid = |\langle \psi||(+)^{\otimes n}\rangle|^2 \) as

\[
fid = \frac{N^{-1}|\sum_k \tilde{f}(x_k)|^2}{(\mathcal{N}_N)^2} = \frac{N^{-2}|\sum_k \tilde{f}(x_k)|^2}{N^{-1}(\mathcal{N}_N)^2} = \frac{N^{-1} \sum_k \tilde{f}(x_k)^2}{N^{-1}(\mathcal{N}_N)^2}.
\]

We can now substitute back our approximations to the function norms as

\[
fid = \frac{\|\tilde{f}\|_1/(b-a)^2}{\|\tilde{f}\|_2^2/(b-a)} + \epsilon = \frac{\|\tilde{f}\|_2^2}{\|f\|_2^2/(b-a)} + \epsilon.
\]
Let us now lower bound this quantity using that generally \( \| \tilde{f} \|_2^2 \leq (b-a) \| f \|_{\text{max}}^2 \) as

\[
fid \geq \frac{\| \tilde{f} \|_2^2}{(b-a)^2 \| f \|_{\text{max}}^2} + \epsilon = F^2 / (b-a)^2 + \epsilon
\]

and we have also substituted back our filling ratio \( F \).

\[ \square \]

The above Lemma guarantees that given any function \( f \) and a sufficiently large resolution \( N \), we can start with the all plus state and efficiently prepare \( \tilde{f} \) with the non-deterministic techniques in Appendix B and Appendix C. The success probability depends on the fidelity which we established above depends polynomially on our filling ratio \( F \). In a second step from \( \tilde{f} \) we can prepare \( f \) efficiently with a success rate at least 1/2. The generalisation to complex functions is straightforward.

3. Asymptotic properties of the rank-one matrix \( A \)

**Lemma 14** (pointwise encoding). Given our pointwise encoding from Eq. (A3) of an arbitrary continuous function \( f \) and the corresponding matrix entries \( A_{kl} := f(x_k)f^*(x_l) \) form Eq. (C1), the matrix norm is constant bounded \( \| A \|_{\text{max}} \leq \| f \|_{\text{max}}^2 \). The only non-zero eigenvalue of \( A/N \) is given by the spectral norm as \( \| A/N \|_2 = \lambda_{\max}^2 / N = (b-a) \| f \|_{\text{max}}^2 + \epsilon \). Here \( \| f \|_2 \) and \( \| f \|_{\text{max}} \) are the square integral (\( L^2 \)) and absolute maximum value norms of the continuous function, respectively. Furthermore, the ratio of matrix norms is generally upper bounded as \( \| A \|_{\text{max}} / \| A/N \|_2 \leq F^{-2} + \epsilon \) by our filling ratio from Definition 3. The error term asymptotically vanishes \( \lim_{N \to \infty} \epsilon = 0 \) and for Lipschitz continuous functions it additionally satisfies \( \epsilon \in \mathcal{O}(N^{-1}) \). See Fig. 7 for a numerical verification of these results.

**Proof.** Let us now define the rank-1 matrix \( A_{kl} := f(x_k)f^*(x_l) \). We can bound the largest entry in the matrix as

\[
\| A \|_{\text{max}} = \max_{k,l} |A_{kl}| = \max_k |f(x_k)f^*(x_l)| = \max_k |f(x_k)|^2 \leq \| f \|_{\text{max}}^2.
\]

The upper bound is actually saturated asymptotically for Lipschitz continuous functions with the error scaling \( \| A \|_{\text{max}} = \| f \|_{\text{max}}^2 + \mathcal{O}(N^{-1}) \). Let us now bound the matrix norm using that \( \| A \|_2 = \| A \|_{H\text{S}} \) for our rank-1 matrix \( A \) as

\[
\| A \|_2^2 = \| A \|_{H\text{S}}^2 = \sum_{k,l} |f(x_k)f(x_l)|^2 = \left( \sum_k |f(x_k)|^2 \right)^2.
\]

We can approximate this expression via the Riemann sum from Lemma 13 as

\[
\| A/N \|_2 = \sum_k |f(x_k)|^2 / N = (b-a) \| f \|_2^2 + \epsilon,
\]

where the error term asymptotically vanishes \( \lim_{N \to \infty} \epsilon = 0 \) and for Lipschitz continuous function it additionally satisfies \( \epsilon \in \mathcal{O}(N^{-1}) \). Finally, we recollect all terms and establish the ratio of matrix norms as

\[
\frac{\| A \|_{\text{max}}}{\| A/N \|_2} = \max_k |f(x_k)|^2 / \sum_k |f(x_k)|^2 / N \leq \| f \|_{\text{max}}^2 / \| f \|_2^2 + \epsilon \leq F^{-2} + \epsilon,
\]

where we have used the general function-norm inequality \( \| f \|_2^2 (b-a) \geq \| f \|_1^2 \). We have also substituted back our filling ratio \( F \) from Definition 3. \[ \square \]

**Lemma 15** (integral encoding). Given an arbitrary continuous efficiently integrable function \( f \) in Definition 3 and our corresponding rank-1 matrix \( A_{kl} := g(x_k)g^*(x_l) \) from Eq. (A3) where \( g(x) \) are non-negative, piecewise integrals of \( f \) via our integral in Eq. (A3), the matrix satisfies \( \| A/N \|_2 = \| f \|_1 \) and its matrix entries are asymptotically constant bounded as \( \| A \|_{\text{max}} = (b-a) \| f \|_{\text{max}} + \epsilon \). Here we have used the \( L^1 \) and absolute maximum norms of the function. Furthermore, the ratio of the matrix norms is generally upper bounded as \( \| A \|_{\text{max}} / \| A/N \|_2 \leq (b-a)F^{-1} + \epsilon \) by our filling ratio from Definition 3. The error term asymptotically vanishes \( \lim_{N \to \infty} \epsilon = 0 \) and for Lipschitz continuous function it additionally satisfies \( \epsilon \in \mathcal{O}(N^{-1}) \). See Fig. 7 for a numerical verification of these results.
Proof. Let us first define the rank-1 matrix with non-negative entries as
\[ A_{kl} := Ng(x_k)g(x_l). \]

We can bound the largest entry in \( A \) as
\[
\|A\|_{\text{max}} = \max_{kl} |A_{kl}| = N \max_k g^2(x_k) = N \max_k f(x_k) \Delta N = (b - a)\|f\|_{\text{max}} + \epsilon,
\]
where we have used from Property that \( g(x_k) = \sqrt{f(x_k)\Delta N + \epsilon} \) and recall that \( \Delta N = (b - a)/N \). The error term asymptotically vanishes \( \lim_{N \to \infty} \epsilon = 0 \) and for Lipschitz continuous function it additionally satisfies \( \epsilon \in O(N^{-1}) \) via Lemma.

Given the matrix \( A \) is rank-1 by definition, it only has 1 non-zero eigenvalue which is equivalent to the norm \( \|A\|_2 \) which we can compute via the (equivalent) Hilbert-Schmidt norm as
\[
\|A\|^2_{HS} = N^2 \sum_{k,l} [g(x_k)g(x_l)]^2 = [N \sum_k g^2(x_k)]^2.
\]

As such, we exactly obtain the largest eigenvalue of the matrix as
\[
\|A/N\|^2_2 = \sum_k g^2(x_k) = \sum_k \int_{x_k}^{x_k+\Delta N} f(x) \, dx = \int_a^b f(x) \, dx = \|f\|_1,
\]
where in the last equation we have used that \( f \) is non-negative and therefore its integral is equivalent to is \( L^1 \) norm.

Finally, we recollect all terms and establish the ratio of matrix norms as
\[
\frac{\|A\|_{\text{max}}}{\|A/N\|_2} = \frac{N \max_k g^2(x_k)}{\sum_k g^2(x_k)} \leq \frac{(b - a)\|f\|_{\text{max}} + \epsilon}{\|f\|_1} + \epsilon = (b - a)\mathcal{F}^{-1} + \epsilon,
\]
where we substituted back our definition of the filling ratio \( \mathcal{F} \) from Definition.

\[ \square \]