

The Efficiency of Feynman's Quantum Computer

Ralph Jason Costales,^{1,*} Ali Gunning,^{1,†} and Tony Dorlas¹

¹*Dublin Institute for Advanced Studies.*

(Dated: September 19, 2023)

Feynman's circuit-to-Hamiltonian construction enables the mapping of a quantum circuit to a time-independent Hamiltonian. Here we investigate the efficiency of Feynman's quantum computer by analysing the time evolution operator $e^{-i\hat{H}t}$ for Feynman's clock Hamiltonian \hat{H} . A general formula is established for the probability, $P_k(t)$, that the desired computation is complete at time t for a quantum computer which executes an arbitrary number k of operations. The optimal stopping time, denoted by τ , is defined as the time of the first local maximum of this probability. We find numerically that there is a linear relationship between this optimal stopping time and the number of operations, $\tau = 0.50k + 2.37$. Theoretically, we corroborate this linear behaviour by showing that at $\tau = \frac{1}{2}k + 1$, $P_k(\tau)$ is approximately maximal. We also establish a relationship between τ and $P_k(\tau)$ in the limit of a large number k of operations. We show analytically that at the maximum, $P_k(\tau)$ behaves like $k^{-2/3}$. This is further proven numerically where we find the inverse cubic root relationship $P_k(\tau) = 6.76 k^{-2/3}$. This is significantly more efficient than paradigmatic models of quantum computation.

I. FEYNMAN'S CLOCK HAMILTONIAN

In the circuit model of quantum computing, a calculation is computed by preparing a set of n logical qubits in the computational basis state $|\Psi_{IN}\rangle = |0_1 0_2 \dots 0_n\rangle$ and acting on this state with a sequence of unitary gates $\hat{U}_1, \hat{U}_2, \dots, \hat{U}_{k-1}, \hat{U}_k$ to reach the output state $|\Psi_{OUT}\rangle = \hat{U}_k \hat{U}_{k-1} \dots \hat{U}_2 \hat{U}_1 |\Psi_{IN}\rangle$. Feynman showed that one can map a quantum circuit to a continuous time evolution with a time-independent Hamiltonian, while still implementing a discrete sequence of unitaries [1], [2]. His solution incorporated the composition of unitary operators outlined below.

Let $\hat{U}_1, \hat{U}_2, \dots, \hat{U}_{k-1}, \hat{U}_k$ be the succession of operations we want to perform on the input state of the n atoms in our 'register' $|\Psi_{IN}\rangle$ as shown in Fig. 1. Adjacent to the register, we add an entirely new set of $k+1$ qubits, called 'program counter sites' which live in a 'clock space' separate from the Hilbert space of the register. The purpose of the clock space is to track the progress of the computation. The program counter can be thought of as an electron tunnelling from one empty site to another. The initial state of the clock space has site 0 of the program counter occupied and the rest unoccupied, which in Dirac notation is given by $|100\dots 0\rangle$. During the computation the program counter evolves to the final state of the clock space $|0\dots 001\rangle$, having only site k occupied, at which stage the computation is complete. It is crucial that the register is observed immediately at this point to ensure that the program counter does not return back down the program line. The complete Hamiltonian, describing the evolution of the register qubits and the clock qubits, is given in (1).

$$\hat{H} = \sum_{i=0}^{k-1} \hat{q}_{i+1}^\dagger \hat{q}_i \hat{U}_{i+1} + \text{H.C.} \quad (1)$$

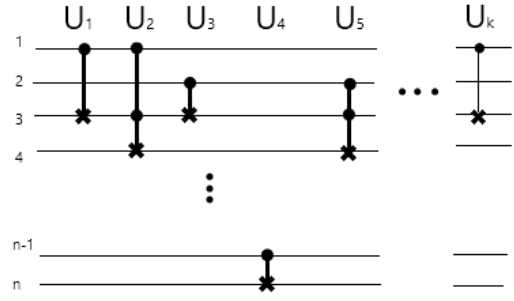


FIG. 1. Schematic of a quantum circuit depicting the succession of \hat{U}_k operators we want to perform on the input state, $|\Psi_{IN}\rangle$, of n atoms in our register to arrive at our desired output state, $|\Psi_{OUT}\rangle$.

Here \hat{q}_i^\dagger and \hat{q}_i are creation and annihilation operators at the i -th clock site, so that $\hat{q}_{i+1}^\dagger \hat{q}_i$ moves the counter one site to the right. Note that the Hermitian conjugate term $\hat{q}_i^\dagger \hat{q}_{i+1}$ occurring in the H.C. term moves the counter one site to the left, so that the clock can also move backwards as time progresses. Our goal is to compute the probability $P_k(t)$ that the computation is complete i.e the probability that the program counter reaches the final state $|0\dots 001\rangle$ at time t for arbitrary number of operations k .

Previously, it was shown by Kitaev et al. [3] and Aharonov et al. [4] that this scheme can be used to implement an arbitrary quantum circuit efficiently using adiabatic evolution i.e. the run-time is polynomial in the number of qubits. The latter authors estimated the spectral gap Δ to be bounded below by $1/(72k^2)$ if k is the number of quantum operations performed. (This bound was improved to $\pi^2/(8k^2)$ in [5].) By the adiabatic theorem (see [6] and [7]) this leads to a running time of the adiabatic algorithm of $O(k^5)$. (According to [6], the dependence on Δ is quadratic. The bound in [7] is more complicated but the example in Section VI suggests that this is still the case in Feynman's Hamiltonian.)

* Presently at Perimeter Institute for Theoretical Physics.

† Presently at Department of Physics, University of Oxford.

Here we consider instead the straightforward evolution of the Feynman Hamiltonian and demonstrate that the computation is complete with probability $O(k^{-2/3})$ in time $\tau = O(k)$. To obtain a success probability of order 1, one therefore has to repeat the calculation $O(k^{2/3})$ times, leading to an estimated run-time of $O(k^{5/3})$. This is a significant improvement over adiabatic evolution. However, this is obviously dependent on whether this can be implemented in practice, as the stopping time τ is very precisely defined, after which there are rapid oscillations of the probability amplitude. Below, in Appendix D, we also estimate the length of the period of these oscillations.

II. ASSESSING THE PROBABILITY OF COMPUTATION COMPLETION

It can be shown that Feynman's Hamiltonian is of the form,

$$\hat{H} = \begin{bmatrix} 0 & U_1^\dagger & & & \\ U_1 & 0 & U_2^\dagger & & \\ & U_2 & 0 & U_3^\dagger & \\ & & \ddots & \ddots & \ddots \\ & & & U_{k-1} & 0 & U_k^\dagger \\ & & & & U_k & 0 \end{bmatrix} \quad (2)$$

The time evolution matrix $\hat{G}(t) = e^{-i\hat{H}t}$ also follows a particular structure given in (3). This can be proven by induction as shown in Appendix B.

$$\hat{G}(t)_{ij} = \begin{cases} a_{ij}(t)\hat{U}_i^\dagger \cdots \hat{U}_{j-1}^\dagger & \text{if } 1 \leq i < j \leq k+1; \\ a_{ii}(t)\hat{\mathbb{I}} & \text{if } 1 \leq i = j \leq k+1; \\ a_{ij}(t)\hat{U}_{i-1} \cdots \hat{U}_j & \text{if } 1 \leq j < i \leq k+1. \end{cases} \quad (3)$$

Given that $\hat{G}(t)$ follows a predictable pattern, the position of the \hat{U}_j 's in the matrix are known for all j . We are only concerned with the $\hat{G}_{(k+1)1}$ element, as it contains the full list of \hat{U}_j 's acting in the required order to obtain the desired computational output state $|\Psi_{OUT}\rangle$. The coefficient $a_{(k+1)1}(t)$ is all we need to find the probability amplitude $P_k(t)$,

$$\begin{aligned} P_k(t) &= |\langle k+1 | \hat{G}(t) | 1 \rangle|^2 \\ &= |a_{(k+1)1}(t)|^2 \end{aligned} \quad (4)$$

$P_k(t)$ here is the desired probability amplitude as it is the time evolution of the clock space from the initial state $|1\rangle$ to the final state $|k+1\rangle$.

The optimal stopping time τ for the register, can be found by maximising $a_{(k+1)1}(t)$. Therefore, $P(t = \tau)$ represents the maximum probability that the computation is complete i.e. all operations have acted on the register.

Given the predictable structure of $\hat{G}(t)$, each \hat{U}_j matrix encoded in the Hamiltonian described in (1) can be ignored. We acquire an effective Hamiltonian that still contains all the information required to obtain the probability amplitude $P_k(\tau)$ as shown in (5).

$$\hat{H}_{\text{eff}} = \begin{bmatrix} 0 & 1 & & & \\ 1 & 0 & 1 & & \\ & 1 & 0 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & 0 & 1 \\ & & & & 1 & 0 \end{bmatrix} \quad (5)$$

This matrix obeys the following eigenvalue equation,

$$\begin{aligned} \hat{H}_{\text{eff}} |\psi_j\rangle &= \lambda_j |\psi_j\rangle \\ e^{-i\hat{H}_{\text{eff}}t} |\psi_j\rangle &= e^{-i\lambda_j t} |\psi_j\rangle \end{aligned}$$

for $j = 1, \dots, k+1$.

Using a Fourier transformation, the eigenvalues and eigenstates of the effective Hamiltonian are found to be,

$$\lambda_j = 2 \cos \frac{\pi j}{k+2} \quad |\psi_j\rangle = \frac{2}{k+2} \begin{bmatrix} \sin \frac{\pi j}{k+2} \\ \sin \frac{2\pi j}{k+2} \\ \vdots \\ \sin \frac{(k+1)\pi j}{k+2} \end{bmatrix} \quad (6)$$

The probability amplitude $P_k(t)$ can now be calculated using (4),

$$P_k(t) = \left| \frac{2}{k+2} \sum_{j=1}^{k+1} e^{-i\lambda_j t} \sin^2 \frac{\pi j}{k+2} (-1)^{j-1} \right|^2 \quad (7)$$

Using (7) for a large number of operations, $k \gg 1$, the optimal stopping time τ is found to be at the first local maximum of $P_k(t)$. As shown in Fig.2 there is a small window to capture the first maximal peak after which there are rapid

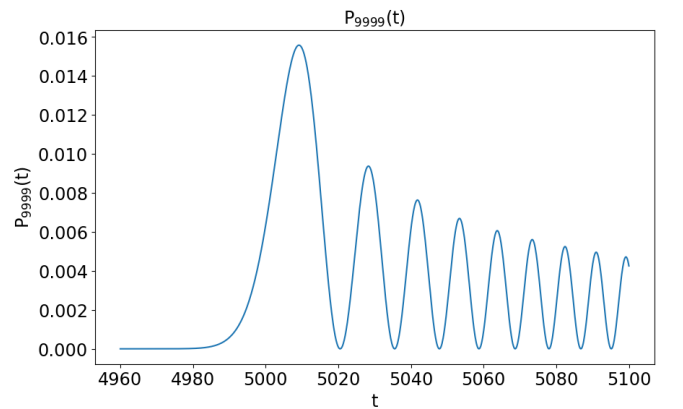


FIG. 2. Plot of $P_k(t)$ for $k=9999$ operators. The first local maximum of $P_k(t)$ is also the global maximum and can be represented by $P_k(\tau)$. The amplitude of each local maxima decreases with time if the program is not immediately stopped at $t = \tau$.

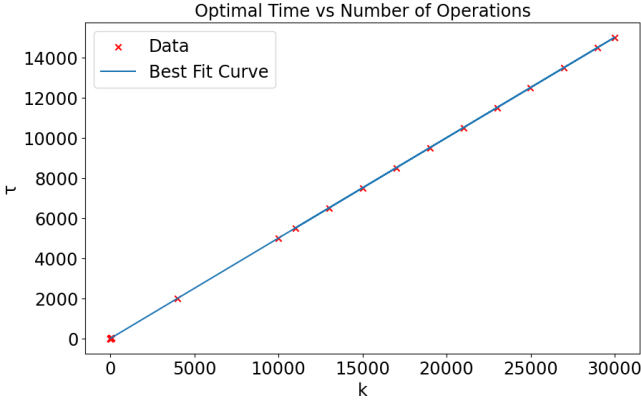


FIG. 3. Scatter plot of the optimal time τ which is found by maximising $P_k(t)$ for a given number of operations k . A linear relationship is observed between τ and k with best fit parameters $\tau = 0.50k + 2.37$.

oscillations of probability which decrease in amplitude with time. Numerically, this optimal stopping time is found to be at $\tau = 0.50k + 2.37$, as shown in Fig.3.

The probability $P_k(\tau)$ can be plotted against k to derive the relationship between probability of computation completion at the optimal time and the number of operations. Numerically, we find that there is an inverse cubic relationship between them, $P_k(\tau) = 6.76k^{-2/3}$, as shown in Fig.4. This relationship is significantly more efficient than paradigmatic models of quantum computation such as those based on adiabatic evolution which were analysed by Aharonov and coworkers [4] (see also [5]).

Analytically, we find a somewhat smaller coefficient. Using Taylor expansions of trigonometric functions the relationship

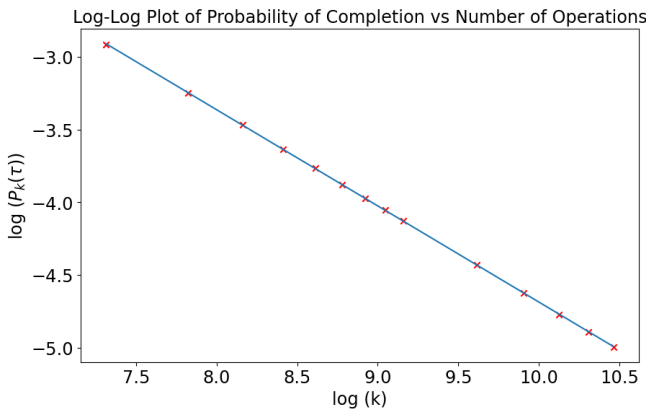


FIG. 4. Log-Log scatter plot of the optimal probability $P_k(\tau)$ which is found by maximising $P_k(t)$ for a given number of operations k . An inverse cubic relationship is observed between $P_k(\tau)$ and k with best fit parameters $P_k(\tau) = 6.76k^{-2/3}$.

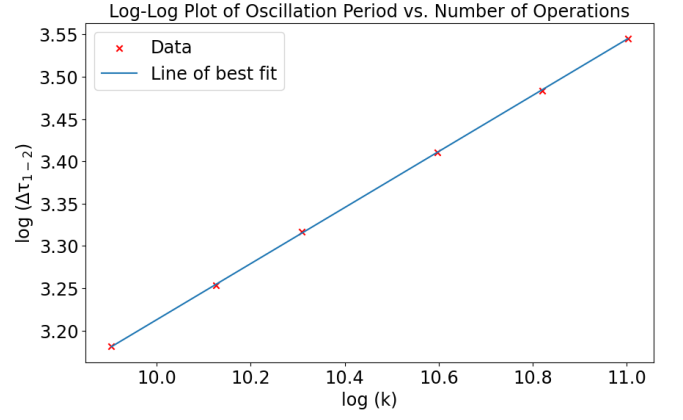


FIG. 5. Log-Log plot of time difference, $\Delta\tau_{1-2}$, between the optimal stopping time, τ_1 , and the next optimal time τ_2 against number of operations, k .

for $P_\tau(k)$ (7) can be transformed into the form,

$$P_k(\tau) = \left| \frac{4}{k+2} \sum_{p=1}^{\frac{k+2}{2}} \cos^2 \left(\frac{\pi(2p-1)}{2(k+2)} \right) \times \cos \left(\frac{\pi^3}{6(k+2)^2} \left(p - \frac{1}{2} \right)^3 \right) \right|^2 \quad (8)$$

This summation form can be approximated by asymptotic analysis, writing it in the form of a Riemann integral assuming that we have a large number of operations, $k \gg 1$: see Appendix C. The result is,

$$P_k(\tau) \approx 5.14k^{-2/3}. \quad (9)$$

It is also important to know the period of the oscillations after the first maximum of $P_k(t)$, as this indicates how accurately one must be able to determine the stopping time τ of the computer (to measure the output state). If $\tau_1 \approx \frac{k+2}{2}$ is the time where the first maximum is attained, we write $\tau_2 = \frac{(k+2)(1+\delta)}{2}$, and $\Delta\tau_{1-2} = \delta \times \frac{k+2}{2}$. Numerically plotting the time difference between the first and second maximum, we find that $\Delta\tau_{1-2} = 1.29(k+2)^{1/3}$. In Appendix D we obtain the analytical prediction $\Delta\tau_{1-2} = 1.115(k+2)^{1/3}$.

This means that one needs to be able to stop the computer at a time accurate up to a fraction of $k^{1/3}$. If the register is not stopped at the optimal time, τ_1 , but is instead stopped at the next optimal time, τ_2 , then the probability $P_k(\tau_2)$ is lower, i.e. it is less likely that the operations have been completed than at τ_1 as can be seen in Fig.2. Nevertheless, it may be more efficient to capture the second maximum rather than restart the program.

III. CONCLUSION

We have presented the theory behind Feynman's theoretical construction for a quantum computer, wherein a quantum

circuit is mapped to a time-independent Hamiltonian by use of a clock space. A formula is established for the probability, $P_k(t)$, that the desired computation is complete at time t for a quantum computer which executes k number of operations. We demonstrate that the computation is complete with probability $O(k^{-2/3})$ in optimal time $\tau = O(k)$. For a success probability of order 1, the calculation must be repeated $O(k^{2/3})$ times, leading to an estimated run-time of $O(k^{5/3})$. This is a significant improvement over adiabatic evolution with a run time $O(k^5)$.

IV. APPENDIX

Appendix A: Feynman's Hamiltonian for $k = 2$

To understand the structure of the Hamiltonian and the time evolution operator, we look at the simplest example of having two operations on our register.

For $k=2$, we add to the n atoms in our register a new set of 3 atoms, which we call 'program counter sites'. q_i and q_i^\dagger represent the annihilation and creation operators, respectively, for the program sites $i = 0, 1, 2$.

Our Hamiltonian in matrix form is given by:

$$\hat{H}_{TOT} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & U_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & U_2^\dagger & 0 & 0 & U_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & U_1 & 0 & 0 \\ 0 & 0 & U_1^\dagger & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & U_1^\dagger & 0 & 0 & U_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & U_2^\dagger & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (A1)$$

This Hamiltonian relates to 2 independent clock spaces. The first is a program counter where at all times only one site is occupied, i.e. $|100\rangle, |010\rangle, |001\rangle$. The second is a program counter where at all times two sites are occupied, i.e. $|011\rangle, |101\rangle, |110\rangle$. The construction of Feynman's Hamiltonian ensures that the number of program sites occupied is a conserved quantity. This allows us to block-diagonalize the Hamiltonian into non-interacting blocks and separate out the part where only one clock site is occupied.

We will suppose that, in the operation of this computer, only one site is occupied for all time. We call this Hamiltonian \hat{H} and can extract it from our total Hamiltonian:

$$\hat{H} = \begin{bmatrix} 0 & U_1^\dagger & 0 \\ U_1 & 0 & U_2^\dagger \\ 0 & U_2 & 0 \end{bmatrix} \quad (A2)$$

This Hamiltonian satisfies the relation

$$\hat{H}^3 = 2\hat{H} \quad (A3)$$

Using a Taylor expansion, the time evolution operator reduces to

$$e^{-i\hat{H}t} = \hat{I} - i\frac{\hat{H}}{\sqrt{2}} \sin \sqrt{2}t + \frac{\hat{H}^2}{2}(\cos \sqrt{2}t - 1) \quad (A4)$$

In matrix form this is

$$e^{-i\hat{H}t} = \begin{bmatrix} 1 + \frac{\cos \sqrt{2}t - 1}{2} & \frac{i}{\sqrt{2}} \sin \sqrt{2}t U_1^\dagger & \frac{\cos \sqrt{2}t - 1}{2} U_1^\dagger U_2^\dagger \\ \frac{-i}{\sqrt{2}} \sin \sqrt{2}t U_1 & \cos \sqrt{2}t & \frac{-i}{\sqrt{2}} \sin \sqrt{2}t U_2^\dagger \\ \frac{\cos \sqrt{2}t - 1}{2} U_2 U_1 & \frac{i}{\sqrt{2}} \sin \sqrt{2}t U_1 & 1 + \frac{\cos \sqrt{2}t - 1}{2} \end{bmatrix} \quad (A5)$$

This matrix is unitary. When the program counter reaches the final site $|001\rangle$ the data register has been multiplied by the operators $U_2 U_1$ as expected. This occurs at $t = \frac{\pi}{3\sqrt{2}}$.

Appendix B: Proving the Structure of a Hamiltonian

To prove the structure given in (3), we expand the time-evolution matrix,

$$G(t) = \sum_{m=0}^{\infty} \frac{(-it)^m}{m!} H^m \quad (B1)$$

Hence it is sufficient to prove that all powers of the Hamiltonian also follow the structure of (3), which will be done via a proof of induction. The first power of the Hamiltonian is, by construction,

$$\hat{H} = \begin{bmatrix} 0 & U_1^\dagger & & & \\ U_1 & 0 & U_2^\dagger & & \\ & U_2 & 0 & U_3^\dagger & \\ & & \ddots & \ddots & \ddots \\ & & & U_{k-1} & 0 & U_k^\dagger \\ & & & & U_k & 0 \end{bmatrix} \quad (B2)$$

Which clearly has the structure in (3). To prove that H^{m+1} has this same structure assuming that H^m does, we approach the problem case by case. Explicitly, each element is given by

$$\begin{aligned} (H^{m+1})_{ij} &= (H H^m)_{ij} \\ &= \sum_l H_{il} H_{lj}^m \\ &= H_{i,i-1} H_{i-1,j}^m + H_{i,i+1} H_{i+1,j}^m \end{aligned}$$

For $i = j$, we get

$$\begin{aligned} (H^{m+1})_{ii} &= a_{i-1,i}^{(m)} U_{i-1} U_{i-1}^\dagger + a_{i+1,i}^{(m)} U_i^\dagger U_i \\ &= (a_{i-1,i}^{(m)} + a_{i+1,i}^{(m)}) \mathbb{I} \end{aligned}$$

For $1 \leq i < j \leq k+1$,

$$(H^{m+1})_{ij} = a_{i-1,j}^{(m)} U_{i-1} U_{i-1}^\dagger U_i^\dagger \dots U_{j-1}^\dagger + a_{i+1,j}^{(m)} U_i^\dagger \dots U_{j-1}^\dagger \\ = (a_{i-1,j}^{(m)} + a_{i+1,j}^{(m)}) U_i^\dagger \dots U_{j-1}^\dagger$$

For $1 \leq j < i \leq k+1$

$$(H^{m+1})_{ij} = a_{i-1,j}^{(m)} U_{i-1} \dots U_j + a_{i+1,j}^{(m)} U_i^\dagger U_i U_{i-1} \dots U_j \\ = (a_{i-1,j}^{(m)} + a_{i+1,j}^{(m)}) U_{i-1} \dots U_j$$

Hence each of power of the Hamiltonian has the same structure as (3), and therefore $G(t)$ must also have the same structure. Note that this derivation also proves that the effective Hamiltonian is given by (5).

Appendix C: Analytic solution of $P_k(\tau)$

Using $\hat{G}(t) = e^{-i\hat{H}t}$ and the eigenvalues and eigenvectors given in (6), an analytic solution for $P_k(t)$ can be found,

$$P_k(t) = |\langle k+1 | \hat{G}(t) | 1 \rangle|^2 \\ = \left| \sum_n \sum_j \langle k+1 | \psi_n \rangle \langle \psi_n | e^{-i\hat{H}_{eff}t} | \psi_j \rangle \langle \psi_j | 1 \rangle \right|^2 \\ = \left| \sum_n \sum_j \langle k+1 | \psi_n \rangle \langle \psi_n | e^{-i\lambda_j t} | \psi_j \rangle \langle \psi_j | 1 \rangle \right|^2 \quad (C1) \\ = \left| \sum_j e^{-i\lambda_j t} \langle k+1 | \psi_j \rangle \langle \psi_j | 1 \rangle \right|^2 \\ = \left| \frac{2}{k+2} \sum_{j=1}^{k+1} e^{-i\lambda_j t} \sin^2 \frac{\pi j}{k+2} (-1)^{j-1} \right|^2$$

In the following, we assume that k is an odd number. The derivation for even k is analogous.

Starting from the expression for $a_{(k+1),1}(t)$ given in (7), we split the sum into two parts,

$$a_{(k+1),1} = \frac{2}{k+2} \sum_{j=1}^{\frac{k+1}{2}} e^{-i\lambda_j t} \sin^2 \frac{\pi j}{k+2} (-1)^{j-1} \\ + \frac{2}{k+2} \sum_{j=\frac{k+1}{2}+1}^{k+1} e^{-i\lambda_j t} \sin^2 \frac{\pi j}{k+2} (-1)^{j-1} \quad (C2)$$

By a change of variable $j = k+2-m$ in the second term,

$$a_{(k+1),1} = \frac{2}{k+2} \sum_{m=1}^{\frac{k+1}{2}} e^{-i\lambda_m t} \sin^2 \frac{\pi m}{k+2} (-1)^{m-1} \\ - \frac{2}{k+2} \sum_{m=1}^{\frac{k+1}{2}} e^{+i\lambda_m t} \sin^2 \frac{\pi m}{k+2} (-1)^{k+1-m} \quad (C3) \\ = \frac{2}{k+2} \sum_{m=1}^{\frac{k+1}{2}} (e^{-i\lambda_m t} - e^{+i\lambda_m t}) \sin^2 \frac{\pi m}{k+2} (-1)^{m-1} \\ = \frac{-4i}{k+2} \sum_{m=1}^{\frac{k+1}{2}} \sin \lambda_m t \sin^2 \frac{\pi m}{k+2} (-1)^{m-1}$$

Substituting in (6) the (approximate) optimal time $\tau = \frac{k+2}{2}$, we get

$$a_{(k+1),1} = \frac{-4i}{k+2} \sum_{m=1}^{\frac{k+1}{2}} \sin \left((k+2) \cos \left(\frac{\pi m}{k+2} \right) \right) \\ \times \sin^2 \frac{\pi m}{k+2} (-1)^{m-1}$$

Using another change of variable, we set $p = \frac{k+1}{2} + 1 - m$,

$$a_{(k+1),1} = \frac{-4i}{k+2} \sum_{p=1}^{\frac{k+1}{2}} \sin \left((k+2) \sin \left(\frac{\pi(2p-1)}{2(k+2)} \right) \right) \\ \times \cos^2 \left(\frac{\pi(2p-1)}{2(k+2)} \right) (-1)^{\frac{k+1}{2}-p}$$

To estimate this sum, we do a second-order Taylor expansion of the argument of the outer sine-function:

$$(k+2) \sin \left(\frac{\pi(2p-1)}{2(k+2)} \right) \approx \left(p - \frac{1}{2} \right) \pi - \frac{\pi^3}{6(k+2)^2} \left(p - \frac{1}{2} \right)^3 \quad (C4)$$

Then noting that $\sin \left(\pi \left(p - \frac{1}{2} \right) - a \right) = (-1)^{p-1} \cos(a)$, we obtain

$$a_{(k+1),1} = \frac{4i(-1)^{\frac{k+1}{2}}}{k+2} \sum_{p=1}^{\frac{k+1}{2}} \cos^2 \left(\frac{\pi(2p-1)}{2(k+2)} \right) (-1)^{\frac{k+1}{2}-p} \\ \times \cos \left(\frac{\pi^3}{6(k+2)^2} \left(p - \frac{1}{2} \right)^3 \right). \quad (C5)$$

This is formula (8). We now note that the second cosine factor is rapidly oscillating for large k . This means that large p -values make a negligible contribution. For small values, the first cosine factor is almost equal to 1 and can be omitted. Introducing the variable $x = \frac{\pi}{(k+2)^{2/3}} \left(p - \frac{1}{2} \right)$, we can approximate the sum by a Riemann integral:

$$a_{(k+1),1} \approx \frac{4i(-1)^{\frac{k+1}{2}}}{\pi(k+2)^{1/3}} \int_0^\infty \cos \left(\frac{1}{6} x^3 \right) dx. \quad (C6)$$

The integral can be evaluated using integration by parts, and we obtain

$$a_{(k+1)1} \approx 2.27i(-1)^{(k+1)/2}(k+1)^{-1/3}. \quad (\text{C7})$$

Squaring this yields (9).

Appendix D: Predicting the Location of the Second Maximum

In Appendix C, we found that the first maximum occurs approximately at $\tau = (k+2)/2$. At this value for t ,

$$2t \left(\frac{(2p-1)\pi}{2(k+2)} - \frac{1}{6} \left(\frac{\pi(2p-1)}{2(k+2)} \right)^3 \right) \approx \left(p - \frac{1}{2} \right) \pi \quad (\text{D1})$$

which is an odd multiple of $\pi/2$. The next maximum should occur when

$$2t \left(\frac{(2p-1)\pi}{2(k+2)} - \frac{1}{6} \left(\frac{\pi(2p-1)}{2(k+2)} \right)^3 \right) \approx \left(p + \frac{1}{2} \right) \pi. \quad (\text{D2})$$

Setting $t = (k+2)(1+\delta)/2$, this becomes

$$\delta \frac{2p-1}{2} - \frac{\pi^2(2p-1)^3}{48(k+2)^2} \approx 1 \quad (\text{D3})$$

On the other hand, we want this to remain a good approximation if we increase or decrease p . This is the case if we set

$$2t \frac{d}{dp} \left(\frac{(2p-1)}{2(k+2)} - \frac{\pi^2}{6} \left(\frac{(p-\frac{1}{2})}{k+2} \right)^3 \right) = 1 \quad (\text{D4})$$

i.e increasing p by 1 also increases the left-hand side by 1. Hence,

$$2t \left(\frac{1}{k+2} - \frac{\pi^2}{2} \frac{(p-\frac{1}{2})^2}{(k+2)^3} \right) = 1. \quad (\text{D5})$$

Inserting $2t = (k+2)(1+\delta)$ we get

$$\delta \approx \frac{\pi^2}{2} \frac{(p-\frac{1}{2})^2}{(k+2)^2}. \quad (\text{D6})$$

Combining this with the previous identity we obtain

$$\frac{\pi^2}{24} \frac{(2p-1)^3}{(k+2)^2} = 1. \quad (\text{D7})$$

Finally inserting this into δ , we get

$$\delta = 0.5(3\pi)^{2/3}(k+2)^{-2/3} = 2.23(k+2)^{-2/3}. \quad (\text{D8})$$

-
- [1] Richard P Feynman, “Quantum mechanical computers,” *Optics news* **11**, 11–20 (1985).
 - [2] Richard P. Feynman and Hey Anthony J G., *Feynman lectures on Computation* (CRC Press, 2023).
 - [3] Alexei Yu Kitaev, Alexander Shen, and Mikhail N Vyalii, *Classical and quantum computation*, 47 (American Mathematical Soc., 2002).
 - [4] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, “Adiabatic quantum computation is equivalent to standard quantum computation,” in *45th Annual IEEE Symposium on Foundations of Computer Science* (2004) pp.

- 42–51.
- [5] Shane Dooley, Graham Kells, Hosho Katsura, and Tony C. Dorlas, “Simulating quantum circuits by adiabatic computation: Improved spectral gap bounds,” *Phys. Rev. A* **101**, 042302 (2020).
- [6] Max Born and Vladimir Fock, “Beweis des adiabatsatzes,” *Zeitschrift für Physik* **51**, 165–180 (1928).
- [7] Sabine Jansen, Mary-Beth Ruskai, and Ruedi Seiler, “Bounds for the adiabatic approximation with applications to quantum computation,” *Journal of Mathematical Physics* **48** (2007).