

Recursive simulation of quantum annealing

A P Sowa¹, M J Everitt², J H Samson², S.E. Savel'ev², A M Zagorskin², S Heidel¹, J C Zúñiga-Anaya³

¹*Department of Mathematics and Statistics, University of Saskatchewan*

²*Department of Physics, Loughborough University, Loughborough, Leics LE11 3TU, UK and*

³*Research Computing Group, ICT, University of Saskatchewan*

(Dated: August 4, 2015)

The evaluation of the performance of adiabatic annealers is hindered by lack of efficient algorithms for simulating their behaviour. We exploit the analyticity of the standard model for the adiabatic quantum process to develop an efficient recursive method for its numerical simulation in case of both unitary and non-unitary evolution. Numerical simulations show distinctly different distributions for the most important figure of merit of adiabatic quantum computing — the success probability — in these two cases.

I. INTRODUCTION

The ongoing debate whether the devices D-Wave One and D-Wave Two, which contain hundreds of superconducting flux qubits, demonstrate quantum or classical annealing [1–8], is to a large extent due to the fact that their quantitative simulation is at the moment impossible. Characteristically, there was no controversy over quantum behaviour of an 8-qubit register [9], where a quantitative agreement was found between the experimental data and the theoretical predictions. Though one cannot expect to be able to model classically a large enough quantum system [10], the development of efficient classical algorithms could hopefully enable us to model quantum systems just big enough to take over. We can also exploit the idiosyncratic features of certain quantum devices, which simplify their modelling by classical means. Here we demonstrate such an algorithm for modelling linear adiabatic evolution, including the case when decoherence is present. This result depends on the analyticity of the final state of the system as a function of evolution time, which is rigorously proven. Our small-scale numerical realizations of this algorithm show that some features of quantum annealing are comparable to the behaviour of D-Wave machines. Since at this stage the simulations do not account for the specific physical conditions that influence the machine's performance we stop short of drawing conclusions as to the strength of signatures of quantumness in the actual device. Nevertheless, we do hope that the tools developed here will enable a better understanding of this important problem.

An adiabatic process evolves the state vector of a quantum annealer over time T along the trajectory $s \mapsto |\psi(s)\rangle$, where the parameter $s \in [0, 1]$ is the reduced time $s = t/T$. The initial value $|\psi(0)\rangle$ is the ground state of the initial Hamiltonian H_i . When the process stops at $t = T$ the state $|\psi(1)\rangle$ is considered as an approximant to the ground state of the custom design Hamiltonian H_f , which encodes the solution of an optimization problem of choice. We consider the standard linear case, when the evolution is driven by the time-dependent Hamiltonian

$$H(s) = (1 - s)H_i + sH_f, \quad (1)$$

according to the Schrödinger equation

$$\frac{d}{ds}|\psi(s)\rangle = -i(T/\hbar)H(s)|\psi(s)\rangle. \quad (2)$$

Let the quantum system be an N -qubit register. In order to fix notation we describe the eigenstates and the corresponding eigenvalues of $H(s)$:

$$H(s)|m; s\rangle = E_m(s)|m; s\rangle, \quad \text{where } E_0(s) \leq E_1(s) \leq \dots \leq E_{2^N-1}(s). \quad (3)$$

However, this assumption is not necessary to derive the analytic results presented in next section. As we will see, our analysis remains valid in every case of bounded (in particular, finite) Hamiltonians H_i, H_f .

The purpose of analysis and simulation of (1) is estimation of the success probability

$$P = \|\Pi\psi(1)\|^2 = \sum_{j=0,1,\dots,d-1} |\langle\psi(1)|j; 1\rangle|^2, \quad (4)$$

where Π denotes the projector onto the ground state space of H_f , e.g. for a non-degenerate ground state $P = |\langle\psi(1)|0; 1\rangle|^2$. (In fact, in most cases of interest to us the dimension of the ground state space d exceeds 1. This is

rigorously accounted for in all numerical results presented in this article.) P provides a measure of the accuracy with which the adiabatic process finds the solution of the underlying problem. For the Hamiltonians considered here, the instantaneous ground state $|0; s\rangle$ is non-degenerate for $0 \leq s < 1$ (see Remarks), although the ground state of H_f may be highly degenerate. In a practical optimisation problem any of these degenerate states would be an equally useful solution. While there is a unique vector in the ground state space adiabatically connected to the initial ground state, this will in general be a superposition of computational basis states (a random one of which will be seen in the readout) and not necessarily of interest in the context of optimisation.

Having fixed T and the number of steps and the Hamiltonian, we repeatedly conduct the numerical experiment. Final Hamiltonians are drawn from an ensemble of binary optimization problems and the simulation outputs the probability of success P for each instance of H_f . We record the outcome of the series of experiments in a histogram of P values. The histogram of the probability of success has long been adopted as a means for testing hypotheses about adiabatic computing, see e.g. [11] and, more recently, [1].

Remarks. It is interesting to briefly consider the evolution of the ground state projector $s \mapsto \Pi_s$, where $\Pi_1 = \Pi$ is as above. Given that in the case at hand the dependence of $H(s)$ on s is linear and *a fortiori* holomorphic, the theory of analytic perturbations applies and it gives us the following picture: A standard general result, see Theorem 1.8, p. 70, [12], tells us that all the eigenvalues $E_k(s)$ as well as the corresponding eigenprojections, including Π_s , will depend on s analytically in the entire complex plane. *A priori* these analytic functions may have algebraic type singularities at a set of isolated exceptional points in the complex plane, where the number of distinct eigenvalues is lower than in the generic case. More precisely, the eigenvalues as functions of s are holomorphic in simply-connected domains that exclude the exceptional points. At the exceptional points an eigenvalue function may have, although does not necessarily have to have, a branching point. Moreover, at a branching point the corresponding eigenprojection will have a pole. *A priori*, such a scenario, i.e. hitting a branching point where the eigenprojection has a singularity, might be construed as a phase transition of the first kind. From this point of view it is not clear *a priori* why a result such as Theorem II.1 should hold. (Note that the result can be reformulated for complex variable s stating that the solution of (2) is an entire function of s .)

If we restrict the variable s to the real line, the picture outlined above is considerably simplified. Indeed, in this case $H(s)$ is a self-adjoint operator, and satisfies the assumptions of Theorem 6.1, p. 120, [12]. This result assures us that both the eigenvalues and the eigenprojections are holomorphic around every point of the real line. In other words, even if some s_0 on the real line is an exceptional point, the projection will not have a pole there and none of the eigenvalue functions will branch. However, we emphasize that the dimension of the ground state may jump discontinuously at the exceptional point, e.g. consider $E_0(s)$ with the corresponding projection Π_s and $E_1(s)$ with $\tilde{\Pi}_s$. It may be that $E_0(s_0) = E_1(s_0)$. Both these projections depend on s holomorphically but the projection onto the ground state of $H(s_0)$ will be $\Pi'_{s_0} = \tilde{\Pi}_{s_0} + \Pi_{s_0}$. This explains why the projections corresponding to the eigenvalues depend on s continuously, while the projection to the ground state may have isolated jump discontinuities. In particular, the latter scenario is bound to play out for at least one point $s_0 \in [0, 1]$ when the ground state spaces of H_i and H_f have different dimensions. At this juncture it is interesting to invoke a recent result of Zhang et al., [13], which shows that in the particular case of Hamiltonians we work with in our numerical simulations, see (29)–(30), the ground state is non-degenerate for all $s \in [0, 1]$. This implies that in all cases for which the ground state of H_f is degenerate the minimal gap tends to zero as s approaches 1 from the left.

II. CONSEQUENCES OF ANALYTICITY

Since Hamiltonian (1) depends on parameter s linearly, the solution of Eq. (2) may be expected to depend on s analytically, at least for small s . Thus, we look for solutions in the form of a Taylor series

$$\psi(s) = \psi_0 + s\psi_1 + s^2\psi_2 + s^3\psi_3 + \dots \quad (5)$$

In fact, we will demonstrate that the radius of convergence of series characterizing solutions of Eq. (2) is infinite, i.e. the series converge for all s , see Theorem II.1 below. In order to simplify notation, let us define auxiliary operators

$$A = -i(T/\hbar) H_i \quad \text{and} \quad B = -i(T/\hbar) (H_f - H_i). \quad (6)$$

Thus, (2) is equivalent to

$$\frac{d}{ds} |\psi(s)\rangle = (A + Bs) |\psi(s)\rangle. \quad (7)$$

Below we will consider this equation for general bounded operators A and B , and only later draw conclusions about the specific case (6) relevant to the application at hand.

Substituting (5) in (7) we readily obtain

$$\begin{aligned} |\psi_0\rangle &= |\psi(0)\rangle \\ |\psi_1\rangle &= A|\psi_0\rangle \\ &\vdots \\ |\psi_n\rangle &= \frac{1}{n} (A|\psi_{n-1}\rangle + B|\psi_{n-2}\rangle) \quad \text{for } n \geq 2 \end{aligned} \tag{8}$$

We use this recurrence as the core of numerical schemas for simulation of the solutions of (1). It allows relatively efficient, as compared to ODE simulation, computation of consecutive terms and consecutive partial sums of the series (5). Numerical experiment shows that this method, implemented directly as a MATLAB script, is typically faster than the standard ODE solver ‘ode45’ by a factor exceeding 3 for $N \leq 12$ qubits. For $N = 14$, in which case the memory constraint plays a significant role, the speed up factor is approximately 60, reducing the computation time from just under one hour to under one minute. Moreover, as described in Section V, a careful implementation of an optimized code allowed us to simulate systems with up to $N = 24$.

A similar procedure can be applied if the interpolation is a polynomial of order m in s : in that case the three-term recurrence relation will be replaced by an $m+2$ -order recurrence. Indeed, this may also be possible for an analytic interpolating function of known Taylor series. However, to demonstrate the method we only discuss the simplest (linear) interpolation here.

Note that evaluation of the success probability (4) requires only the computation of

$$P = |\Pi(\psi_0 + \psi_1 + \psi_2 + \dots)|^2.$$

In other words, the solution is found, immediately as it were, at the point $s = 1$ without the need of finding all the intermediate states $|\psi(s)\rangle$. This is in stark contrast to the computation based on an application of an ODE solver to Eq. (2). The current method has been tested against our earlier study of the correlation between gap and success probability[14]; in that work we used a conventional (Dormand-Prince) solver. Note also that if series (5) is known to converge absolutely, then $|\psi(s)\rangle$ automatically satisfies (2). Therefore, the main issue at stake is the estimation of the radius of convergence

$$R_c = \left(\limsup_{n \rightarrow \infty} \|\psi_n\|^{1/n} \right)^{-1}. \tag{9}$$

Here $\|\cdot\|$ denotes the ℓ_2 norm. In general R_c depends on the constituents of the process H_i, H_f and possibly even $|\psi(0)\rangle$. From the point of view of simulations it is ideal to have $R_c = \infty$, which ensures that $|\psi(s)\rangle$ given in (5) is the solution of (2) for all (reduced) times $s = t/T$. If on the other hand, $R_c < 1$, then the series cannot be used to estimate the success probability. But, as we shall see, the series (5) converges everywhere.

Let us introduce the following notation:

$$a := \|A\|, \quad b := \|B\|. \tag{10}$$

Here, $\|\cdot\|$ is the operator norm. Throughout the article we assume $0 < a, b < \infty$. For the particular case of (6), we have $a = (T/\hbar)\|H_i\|$, $b = (T/\hbar)\|H_i - H_f\|$. The Hamiltonians are nontrivial and bounded, e.g. finite dimensional, and, moreover, $H_i \neq H_f$. We have the following result:

Theorem II.1. $R_c = \infty$, i.e. series (5) converges absolutely in the entire complex plane and its limit $\psi(s)$ for $s \in [0, \infty)$ is the solution of (7) satisfying the initial condition $\psi(0) = \psi_0$.

Proof. Recurrence (8) readily implies

$$|\psi_n\rangle = \frac{1}{n!} P_n |\psi_0\rangle, \tag{11}$$

where $P_n = P_n(A, B)$ are operators defined via recurrence

$$\begin{aligned} P_0 &= I \\ P_1 &= A \\ &\vdots \\ P_{n+1} &= AP_n + nBP_{n-1} \quad \text{for } n \geq 1. \end{aligned} \tag{12}$$

Note that we have $\|P_0\| = 1$, $\|P_1\| = a$, and $\|P_{n+1}\| = \|AP_n + nBP_{n-1}\| \leq a\|P_n\| + nb\|P_{n-1}\|$. We wish to estimate the rate of growth of $\|P_n\|$. To this end let us consider an auxiliary scalar sequence (p_n) defined via recurrence

$$\begin{aligned} p_0 &= 1 \\ p_1 &= a \\ &\vdots \\ p_{n+1} &= ap_n + nbp_{n-1} \quad \text{for } n \geq 1. \end{aligned} \tag{13}$$

Since $p_0 = \|P_0\|$, $p_1 = \|P_1\|$ and p_n grows at least as fast as $\|P_n\|$, we clearly have

$$\|P_n\| \leq p_n. \tag{14}$$

Next, we undertake to estimate the growth rate of (p_n) . First, observe that $p_n = p_n(a, b)$ may be viewed as a polynomial in two variables, e.g. $p_2 = a^2 + b$, $p_3 = a^3 + 3ab$, etc. It is easily seen that the general form of the polynomials is

$$p_n = c_0[n]a^n + c_1[n]a^{n-2}b + c_2[n]a^{n-4}b^2 + c_3[n]a^{n-6}b^3 + \dots = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} c_k[n]a^{n-2k}b^k, \tag{15}$$

where the coefficients $c_k[n]$ remain to be found. Note that the last term of the polynomial is either $c_{n/2}[n]b^{n/2}$ (when n is even) or $c_{\lfloor n/2 \rfloor}[n]ab^{\lfloor n/2 \rfloor}$ (when n is odd), with $\lfloor x \rfloor$ denoting the integer part of x . Moreover, it is easily seen that (13) implies

$$\begin{aligned} c_0[n] &= 1 \\ c_1[n] &= \binom{n}{2} \\ &\vdots \\ c_k[n] &= c_k[n-1] + (n-1)c_{k-1}[n-2] \end{aligned} \tag{16}$$

Using this, and applying induction, one readily obtains an explicit formula

$$c_k[n] = (2k-1)!! \binom{n}{2k}, \tag{17}$$

where $(2k-1)!! = 1 \cdot 3 \cdot 5 \cdot \dots \cdot (2k-1)$. In light of this (15) yields

$$\frac{1}{n!} p_n = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \frac{1}{k!(n-2k)!} a^{n-2k} \frac{b^k}{2^k}. \tag{18}$$

Next, we make the following observation

$$k!(n-2k)! \geq \left\lfloor \frac{n}{3} \right\rfloor! \quad \text{for } k = 0, 1, 2, \dots, \lfloor n/2 \rfloor. \tag{19}$$

Indeed, we either have $k \geq \lfloor \frac{n}{3} \rfloor$ or else $n-2k > n-2\lfloor \frac{n}{3} \rfloor \geq \lfloor \frac{n}{3} \rfloor$. In either case (19) follows trivially.

Next, we obtain from (18) and (19):

$$\begin{aligned} \frac{1}{n!} p_n &= \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \frac{1}{k!(n-2k)!} a^{n-2k} \frac{b^k}{2^k} \\ &\leq \frac{1}{\left\lfloor \frac{n}{3} \right\rfloor!} a^n \left\{ 1 + \frac{b}{2a^2} + \left(\frac{b}{2a^2}\right)^2 + \left(\frac{b}{2a^2}\right)^3 + \dots + \left(\frac{b}{2a^2}\right)^{\lfloor n/2 \rfloor} \right\} \\ &\leq \frac{1}{\left\lfloor \frac{n}{3} \right\rfloor!} a^n \left(1 + \frac{b}{2a^2} \right)^{n/2}. \end{aligned} \tag{20}$$

As is well known, $\left(\left\lfloor \frac{n}{3} \right\rfloor!\right)^{1/n} \rightarrow \infty$ as $n \rightarrow \infty$. Thus, recalling definition (11), we obtain

$$\|\psi_n\|^{1/n} \leq \left\| \frac{1}{n!} P_n \right\|^{1/n} \|\psi_0\|^{1/n} \leq \left(\frac{1}{n!} p_n \right)^{1/n} \|\psi_0\|^{1/n} = \frac{1}{\left(\left\lfloor \frac{n}{3} \right\rfloor!\right)^{1/n}} a \left(1 + \frac{b}{2a^2} \right)^{1/2} \|\psi_0\|^{1/n} \rightarrow 0, \tag{21}$$

which by (9) implies $R_c = \infty$. \square

Remark 1. Note that when $b < 2a^2$ (valid in the adiabatic regime) estimate (20) may be replaced by a more efficient one. Indeed in such a case

$$1 + \frac{b}{2a^2} + \left(\frac{b}{2a^2}\right)^2 + \left(\frac{b}{2a^2}\right)^3 + \dots \leq \frac{1}{1 - \frac{b}{2a^2}}$$

If in addition $a < 1$, then terms $p_n/n!$ diminish very fast which ensures fast convergence of series (5) when $s = 1$, also in the numerical sense. However, this is not of interest in the current problem as it represents the anti-adiabatic limit, as the number of spin flips during the evolution is of order a . In the special case of interest $a = (T/\hbar)\|H_i\|, b = (T/\hbar)\|H_i - H_f\|$ and the condition $a < 1$ & $b < 2a^2$ is equivalent to

$$\hbar \frac{\|H_i - H_f\|}{2\|H_i\|^2} < T < \hbar \frac{1}{\|H_i\|}, \text{ which implies } \|H_i - H_f\| < 2\|H_i\|.$$

This assumption on T ensures the most efficient computation of series (5). However, we do not claim that this restriction delineates the only regimes in which computation is effective.

III. THE MASTER EQUATION

Now we will extend the above results to the evolution described by the master equation (see e.g. [15]):

$$\frac{d}{dt}\rho = -\frac{i}{\hbar}[H, \rho] + L\rho L^\dagger - \frac{1}{2}L^\dagger L\rho - \frac{1}{2}\rho L^\dagger L.$$

Here, ρ is the density matrix, H is the quantum system Hamiltonian and the Lindblad operator L accounts for decoherence due to the uncontrollable interaction of the quantum system with the environment. In some applications the master equation involves a finite number of different Lindblad operators. While for our purposes it is sufficient to have just one Lindblad, the results presented here can easily be extended to the more general case. From now on we will only consider the evolution of finite-dimensional systems. Furthermore, as before we assume that H takes the special form (1) with $s = t/T$. Substituting variables (6) as before, we obtain an equivalent form of the master equation

$$\frac{d}{ds}\rho = \mathcal{L}_A[\rho] + s\mathcal{L}_B[\rho] \tag{22}$$

with the shorthand notation

$$\mathcal{L}_A[\rho] := [A, \rho] + T\left(L\rho L^\dagger - \frac{1}{2}L^\dagger L\rho - \frac{1}{2}\rho L^\dagger L\right), \quad \text{and} \quad \mathcal{L}_B[\rho] := [B, \rho]. \tag{23}$$

Note that the evolution equation (22) generalizes the von Neumann equation

$$\frac{d}{ds}\rho(s) = -i(T/\hbar)[H(s), \rho(s)]. \tag{24}$$

In the context of adiabatic computing, it is vital to realize that there is an essential difference between the two types of evolution. Namely, at least if certain general conditions on H_f are satisfied, the adiabatic theorem applies to the case (24) and ensures that the probability of success increases with increasing T and will tend to one as T tends to infinity. In contrast, there is no known generalization of the adiabatic theorem, extending it to the case of (22). In fact, in the course of numerical experimentations we have observed that for a relatively large L the probability of success may decrease with increasing time T . Figure 3 illustrates this phenomenon. This observation agrees with the conclusions of Ref. [16] that there is a finite time window for the operation of a quantum annealer interacting with the thermal bath.

In order to describe the analytic properties of (22) we introduce the Hilbert space structure on the linear space of $K \times K$ matrices (e.g. $K = 2^N$ for an N qubit system). For $X \in \mathbb{C}^{K \times K}$ we define the Hilbert-Schmidt norm $\|X\|_{HS} = \text{Tr}(XX^\dagger)^{1/2}$. This norm endows $\mathbb{C}^{K \times K}$ with the Hilbert space structure. We denote the corresponding operator norms of $\mathcal{L}_A, \mathcal{L}_B : \mathbb{C}^{K \times K} \rightarrow \mathbb{C}^{K \times K}$ simply by $\|\mathcal{L}_A\|, \|\mathcal{L}_B\|$. As is well known, the Hilbert-Schmidt norm

has the submultiplicative property $\|AB\|_{HS} \leq \|A\|_{HS} \|B\|_{HS}$ which, when applied to (23), yields the following estimates:

$$\|\mathcal{L}_A\| \leq 2\|A\|_{HS} + 2T\|L\|_{HS}^2, \quad \|\mathcal{L}_B\| \leq 2\|B\|_{HS}. \quad (25)$$

Next, we wish to consider solutions of (22). To this end we introduce the Taylor series Ansatz

$$\rho(s) = \rho_0 + s\rho_1 + s^2\rho_2 + s^3\rho_3 + \dots \quad (26)$$

As before, we substitute (26) into (22). In this way we obtain the following description of the Taylor series coefficients:

$$\rho_n = \frac{1}{n!} Q_n(\mathcal{L}_A, \mathcal{L}_B)\rho_0, \quad (27)$$

where $Q_n = Q_n(\mathcal{L}_A, \mathcal{L}_B)$ is a polynomial in variables \mathcal{L}_A and \mathcal{L}_B defined by recursion:

$$\begin{aligned} Q_0 &= I \\ Q_1 &= \mathcal{L}_A \\ &\vdots \\ Q_{n+1} &= \mathcal{L}_A Q_n + n\mathcal{L}_B Q_{n-1} \quad \text{for } n \geq 1. \end{aligned} \quad (28)$$

We observe that the proof of Theorem II.1 can be repeated almost verbatim with obvious modifications such as: replacing A with \mathcal{L}_A , B with \mathcal{L}_B , and P_n with Q_n , and redefining $a := \|\mathcal{L}_A\|$ and $b := \|\mathcal{L}_B\|$. This yields the following result:

Theorem III.1. *Series (26) with coefficients defined via (27) and (28) converges absolutely in the entire complex plane. Moreover, its limit $\rho(s)$ for $s \in [0, \infty)$ is the solution of (22) satisfying the initial condition $\rho(0) = \rho_0$.*

Remark 2. Consider the unitary map $U(s)$ defined by the adiabatic Schrödinger equation:

$$\frac{d}{ds}U(s) = -i(T/\hbar)H(s)U(s), \quad \text{so that } U(s)|\psi(0)\rangle = |\psi(s)\rangle.$$

Note that recurrence (12) applied in the special case — i.e. $A = -i(T/\hbar)H_i$, $B = -i(T/\hbar)(H_f - H_i)$ — provides a constructive description or simulation of $U(s)$ via the formula

$$U(s) = \sum_{n=0}^{\infty} \frac{1}{n!} P_n.$$

In particular this description of the semigroup $s \mapsto U(s)$ may be used to simulate the evolution of mixed states (24). Indeed,

$$\frac{d}{ds}\rho(s) = -i(T/\hbar)[H(s), \rho(s)] \implies \rho(s) = U(s)\rho(0)U(s)^*.$$

IV. ALGORITHM ANALYSIS

We will utilize elements of the proof of Theorem II.1 to analyze the efficiency of the recurrence algorithm (8) with the following specific problem constituents:

$$H_i = -\sum_k \sigma_k^x, \quad k \in 1, 2, \dots, N, \quad (29)$$

$$H_f = H_{\text{Ising}} = -\sum_{k<l} J_{kl} \sigma_k^z \sigma_l^z, \quad k, l \in 1, 2, \dots, N. \quad (30)$$

Here, $J_{ij} \in \{-1, 1\}$ denote random variables with uniform distribution $P(J_{ij} = 1) = P(J_{ij} = -1) = 1/2$, and N denotes the number of qubits.

First, in order to estimate the efficiency of a recurrence one needs to establish a stopping condition. It is desirable to stop recurrence when the approximation error is sufficiently small. However, little is known a priori about the solution that is sought in (8) and so the calculus methods do not seem to provide an easy estimate for the dependence of error on the degree of approximation n . As a substitute we propose the following stopping condition:

$$\text{Compute } \Psi_k \text{ for } k = 1, 2, \dots, n, \text{ where } n \text{ is the smallest index such that } \|\Psi_n\|^{1/n} \leq \varepsilon. \quad (31)$$

Here, ε is regarded as a preset measure of accuracy. In light of the proof of Theorem II.1 we know that n defined by (31) is bound to be finite. Moreover, since the solution $\Psi(s)$ is an analytic function in the complex plane this is expected to provide a good estimate of the remainder error when s is bounded, e.g. $s \sim 1$.

Next, consider the particular case $T = 1$ (recall also $\hbar = 1$). Let us assume that the Taylor series is centred at s_0 . In such a case $a = \|H_i + s_0(H_i - H_f)\| = O(\|H_f\|) = O(N^2)$, $b = \|H_i - H_f\| = O(\|H_f\|) = O(N^2)$ which readily implies

$$a \left(1 + \frac{b}{2a^2}\right)^{1/2} = \left(a^2 + \frac{1}{2}b\right)^{1/2} = O(N^2). \quad (32)$$

It now follows from (21), (31) and the Stirling formula that

$$a \left(1 + \frac{b}{2a^2}\right)^{1/2} \sim \varepsilon \left(\left[\frac{n}{3}\right]!\right)^{1/n} \sim \varepsilon n^{1/3}, \quad \text{i.e. } n \sim \frac{1}{\varepsilon^3} O(N^6)$$

Now, let us consider $T \gg 1$. While Theorems II.1 and III.1 ensure that the Taylor series defining solutions of Eqns. (7) and (22) converge in the entire complex plane, due to the roundoff errors, the Taylor series need not converge for relatively large values of T as intermediate sums become exponentially large in the adiabatic regime. The largest term $x^n/n!$ in the Taylor expansion of e^x occurs when $n \approx |x|$ and is of the order of $e^{|x|}$. To obtain solutions in such cases we use the following observation. Defining $\tau = s - s_0$ we observe that $\psi(\tau)$ satisfies (7) where s has been replaced by τ and A replaced by $A + s_0B$ with the initial condition $\psi|_{\tau=0} = \psi(s_0)$. Similarly, $\rho(\tau)$ satisfies (22) where s has been replaced by τ and A replaced by $A + s_0B$ with the initial condition $\rho|_{\tau=0} = \rho(s_0)$. This enables one to conduct computation in stages by partitioning the s -interval $[0, 1]$ into smaller segments or subintervals. Practice indicates that the best course of action is to carry out the computation in $\sim T$ subintervals. Since computation within each segment is based on a faster converging Taylor series we obtain convergence of the solution in the entire s -interval for larger values of T .

Note that the estimate (32) holds for every subinterval, which ensures that every segment requires the computation of $O(N^6)$ coefficients. Finally, it is easily seen that the number of arithmetical operations necessary to find n Taylor coefficients is of rank $O(n2^N)$. Thus, the overall number of arithmetical operations required to carry out the computation is of rank $O(T \cdot n \cdot 2^N) = O(T \cdot N^6 \cdot 2^N) = O(2^N)$. This is a polynomial time algorithm when measured against the classical problem size 2^N . Naturally, and not very surprisingly, it is exponential in the number of qubits. The important feature of this algorithm is that it scales linearly with time T .

As an example, we consider the Landau-Zener Hamiltonian $H(s) = (1 - 2s)\sigma_z + \Delta\sigma_x$. It is easily seen that its minimal gap is given by $E_1(s) - E_0(s) = 2\sqrt{\Delta^2 + (1 - 2s)^2}$, with the minimum occurring at $s = .5$. Consider the solution $\Psi(s)$ of (2) with $\Psi(0)$ prescribed as the ground state of $H(0)$. For the sake of an example let us attempt to compute $\Psi(1)$ for $\Delta = 1, T/\hbar = 20$. In this instance the Taylor polynomial centered at $s = 0$ will not give meaningful results: experiment shows that the algorithm outputs $\Psi(1) = [-0.0455 - 0.1663i, -0.4112 + 1.1521i] \cdot 10^{11}$ after 100 iterations. Moreover one does not obtain correct convergence by simply increasing the number of iterations. Indeed, the norm of the simulated $\Psi(1)$ will converge to the value of ~ 1.05 instead of the correct $\|\Psi(1)\| = 1$. This is due to rounding error in the large intermediate terms. Indeed, all computations are done in floating point arithmetic with precision in the order of 10^{-16} . This means that the difference between any two consecutive real numbers in the order of 10^{11} is as large as 10^{-5} , which seriously impedes computational precision. In such a case a split of the interval into two parts, $0 \leq s \leq 0.5$ and $0.5 \leq s \leq 1$, results in smaller intermediate values and ensures convergence to the correct limit. This may be verified by a nearly exact calculation obtained via an application of, say, 200, instalments, see Fig. 11. The figure also shows the expected nonmonotonic convergence to unity with increasing T , with oscillations determined by the time-averaged Larmor frequency. In addition, we note that a symbolic implementation of our method may be used to reproduce the Taylor series of the special function solution. However, numerical evaluation of special functions is frequently quite problematic, and so the calculation via instalments offers a more practicable approach even if the special function solutions were known in closed-form.

Finally, we have used this specific example to benchmark the accuracy of our numerical method (termed TPT for Target-Point-Taylor) by comparing it with the standard fourth-order Runge-Kutta method (RK or ‘ode45’), [17]. Based on a simple test we have observed the following: The TPT method gives the value of

$$\Psi(1) \simeq [0.509629891598850 + 0.766898007985489 i, -0.226356412675608 - 0.317659555887512 i],$$

and

$$P \simeq 0.999801214304354$$

with just two steps computing 350 Taylor coefficients at each step. Re-computation based on 200 intermediate steps shows that the value of $\Psi(1)$ is accurate to 10 significant digits while the value of P is accurate to 9 significant digits. Secondly, we have attempted to reach maximal accuracy via the RK method. In order to obtain comparable accuracy (7 significant digits for $\Psi(1)$ and 9 significant digits for P), namely

$$\Psi(1) \simeq [0.509629851149443 + 0.766898035396728 i, -0.226356395949650 - 0.317659566414614 i],$$

and

$$P \simeq 0.999801214234416$$

we were forced to use a very fine time-discretization $ds = 1.86 \times 10^{-9}$ as larger time-steps lead to greater error. This example suggests that for comparable accuracy the ratio of the number of steps required by RK to those required by TPT exceeds $5.4 \times 10^8 : 700$. As mentioned above, the ODE-solver based methods are impracticable for larger systems due to long computation time.

Remark 3. While the formal analysis above suggests that the constant of proportionality between the number of Taylor coefficients n and N^6 may be very large ($\sim \varepsilon^{-3}$), numerical evidence suggests that a rigid *a priori* number, say $n = 50 \dots 90$ is typically sufficient to ensure numerical convergence. However, for a small fraction of a percentage of Ising Hamiltonians there is no convergence under these conditions. Nevertheless, in the type of experiment considered in this article those cases are statistically insignificant and may be discarded.

Remark 4. We note that the recurrence algorithm does not require that all the Taylor series coefficient terms be kept in the memory. Indeed, in the case of unitary evolution with the dynamic variable being a vector Ψ of length 2^N , the memory requirement is of rank $3 \cdot 2^N$, regardless of the number of coefficients n that are kept in the approximation. Indeed, we only need to keep in the memory the variable Ψ , which is continually updated, and the two last Taylor coefficients, Ψ_k and Ψ_{k-1} as k runs from 2 to n . For a similar reason the memory requirement in the non-unitary case, when the dynamic variable is ρ of size 2^{2N} , is of rank $3 \cdot 2^{2N}$.

V. NUMERICAL EXPERIMENTS

We have conducted numerical experiments for both a pure adiabatic process (2) and the process with dissipation (22) using matrix Hamiltonian as in (29) and (30). The choice of values for J_{ij} ’s is generated via MATLAB’s generic random number generators. It is done independently for every run of the experiments discussed below. Note that $\|H_i\| = N$ and $\|H_f\| \leq N(N-1)/2$.

Following the discussion on Section IV, we divide the s-interval in $\sim T$ subintervals. Even though convergence in each subinterval is faster, the computational time is generally increased by a constant factor not exceeding the number of stages.

The numerical simulations were implemented using compiled MATLAB code. The running time was lessened by using sparse matrices and refactoring the formulas (6) and (8) to reduce the number of matrix-vector calculations in favour of vector-vector calculations. The construction of H_i and H_f was made much more efficient by exploiting the structure and symmetries of the matrices. Memory requirements were also lessened by taking advantage of the sparseness and symmetry in the matrix computations. As a result, N up to 24 becomes computable on a modest distributed-computing server.

Appendix II provides the source code of our current MATLAB implementation of the algorithm. The `Main` function executes the simulation the indicated number of times, using a sparse matrix `H_i` (the Heisenberg matrix representation of the initial Hamiltonian) and a different sparse vector `H_f` (the diagonal of the Random Ising Hamiltonian in the Heisenberg matrix representation) at each time. Functions `InitialHamiltonian` and `RandomIsing`

create those sparse structures using fractal-like iterations. Finally, function `Taylor_Installments` calculates the Taylor series.

The histograms presented in Appendix I were produced using up to 180 cores of 2.0GHz Intel E5-2640L Xeon processors distributed on the compute cluster PLATO of the University of Saskatchewan. To illustrate the performance of the implemented algorithm, a total of 21780 individual simulations were ran for $N = 20$ in about 14 hours, and using less than 5GB of memory per simulation.

Figures 9 and 10 below illustrate the evolution of the compute time per simulation as a function of the number of qubits. An exponential increase of the compute time for large numbers of qubits can be observed. We expect this to be the case for any number of qubits and, in consequence, compute times for simulations of larger systems can be estimated. The fact that this exponential tendency is not verified when the number of qubits is too small, is certainly due to the fact that MATLAB, and the server itself, requires some time for internal processes (other than the execution of the algorithm). It appears that this time is much larger than the execution time of the algorithm for these cases when the number of qubits is too small.

The crucial limiting factor in simulation of larger systems is memory capacity. However, early stage evidence shows that further algorithm optimizations, specialized to 3TB RAM server architectures, enables simulations of size up to 36 qubits. A detailed account of this ongoing work goes outside the scope of this article and will be reported elsewhere.

A. Unitary adiabatic evolution: Schrödinger's equation

We use a schema based on recurrence (8). Furthermore, we set $|\psi_0\rangle$ to be the ground state of H_i . We note that H_f is exponentially degenerate: the Hilbert space is 2^N -dimensional but there are only $O(N)$ distinct eigenvalues, which are separated by multiples of 2. In particular, the Hamiltonian $H(s)$ is invariant under a global spin flip for all time, so adiabatic evolution occurs in the subspace spanned by equal superpositions of computational states and the corresponding spin-flipped states, $\{(|abc\dots\rangle + |\bar{a}\bar{b}\bar{c}\dots\rangle)/\sqrt{2}\}$. Let Π denote the projector onto the ground state space of H_f . (Note that Π is computed numerically, which presents no difficulties.) The probability of success is defined as $P = \|\Pi\psi(1)\|^2$.

The results from simulation are presented via probability of success histograms, Fig. 1, and Figs. 4–8. Of note is the ragged landscape feature of the histograms. This can be understood qualitatively in the following way. The success probability is largely determined by the low-lying eigenvalue structure. The aforementioned degeneracy reduces the number of qualitatively different classes and we conjecture that the peaks correspond to different classes. We conjecture a connection between the low-P tail of the distribution of probability of success and the existence of optimization problems that are inherently hard. Since the optimization problem at hand is known to be NP hard, [18], we expect this phenomenon to persist for larger systems.

B. Non-unitary adiabatic evolution: Master equation with one Lindblad operator

In this setting the probability of success is defined as $P = \text{Tr}(\Pi\rho(1))$ where, again, Π is the projector onto the ground state space of H_f . In order to compute $\rho(1)$ we use a schema based on recurrence (in notation of Subsection III):

$$\begin{aligned} \rho_0 &= |\psi_0\rangle\langle\psi_0|, \quad \text{where } \psi_0 \text{ denotes the ground state of } H_i \\ \rho_1 &= \mathcal{L}_A\rho_0 \\ &\vdots \\ \rho_{n+1} &= \frac{1}{n+1}\mathcal{L}_A(\rho_n + \mathcal{L}_B\rho_{n-1}) \quad \text{for } n \geq 1. \end{aligned} \tag{33}$$

For the particular experiments presented in Figure 2 we took $L = 0.1\hat{a}$ where \hat{a} is the operator, which in the eigenbasis of H_f assumes the off-diagonal form

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & \sqrt{2} & 0 & 0 & \dots \\ 0 & 0 & 0 & \sqrt{3} & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{34}$$

The basis states are in order of increasing energy, with basis states in the degenerate subspaces listed in a particular order. The effect of this H_f -dependent Lindblad operator, in isolation, would be to relax the system to the selected ground state, thereby breaking the degeneracy. However, since it has a highly non-local action on the qubits, its interaction with $H(s)$ disturbs the adiabatic evolution and destroys the fine structure of the probability histogram, see Fig. 3.

Acknowledgements. APS thanks the Department of Physics, Loughborough University, for hosting him in the autumn of 2013 and acknowledges partial support of the Canadian Foundation for Innovation, grant LOF # 22117. The authors are grateful to the referees for pointing out ways to improve the presentation of this material.

-
- [1] S. Boixo, T. F. Rønnow, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, and M. Troyer, *Nature Physics* **10**, 218 (2014)
 - [2] T.F. Rønnow et al., arXiv:1401.2910 (2014)
 - [3] Z. Bian et al., *Phys. Rev. Lett.* **111**, 130505 (2013)
 - [4] J.A. Smolin and G. Smith, arXiv:1305.4904 (2013)
 - [5] L. Wang et al., arXiv:1305.5837 (2013)
 - [6] S.W. Shin et al., arXiv:1401.7087 (2014)
 - [7] P.J.D. Crowley et al., *Phys. Rev. A* **90**, 042317 (2014)
 - [8] M.H. Amin, arXiv:1503.04216 (2015)
 - [9] M.W. Johnson et al., *Nature* **473**, 194 (2011)
 - [10] R. Feynman, *Int. J. Theor. Phys.* **21**, 467 (1982)
 - [11] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, D. Preda, *SCIENCE* **292** (2001), 472–476
 - [12] T. Kato, *Perturbation Theory for Linear Operators*, Springer-Verlag, Berlin, Heidelberg, New York, 1980
 - [13] D-J. Zhang, X-D. Yu, and D. M. Tong, *Phys. Rev. A* **90**, 042321 (2014)
 - [14] M Cullimore, M J Everitt, M A Ormerod, J H Samson, R D Wilson and A M Zagoskin, *J. Phys. A: Math. Theor.* **45** 505305 (2012)
 - [15] I. Percival, *Quantum State Diffusion*, Cambridge University Press, Cambridge, UK, 1998
 - [16] M.S. Sarandy and D.A. Lidar, *Phys. Rev. Lett.* **95**, 250503 (2005)
 - [17] Based on code provided by M. Cullimore.
 - [18] F. Barahona, *J. Phys. A: Math. Theor.* **15** (1982), 3241–3253

Appendix I: Figures

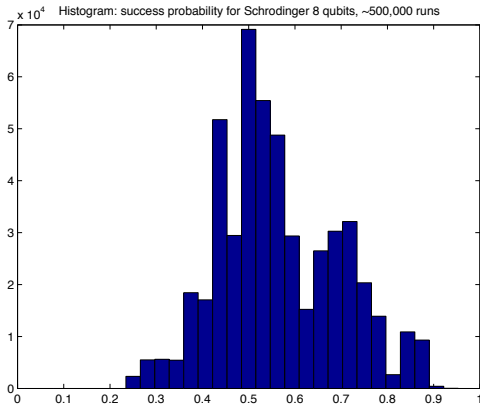


Figure 1. Results obtained via computation of Taylor series solutions of the 8 qubit adiabatic Schrödinger equation with $T = 4$. The histogram displays probability of success outcomes for 500,000 random Ising Hamiltonians H_f sorted into 32 bins.

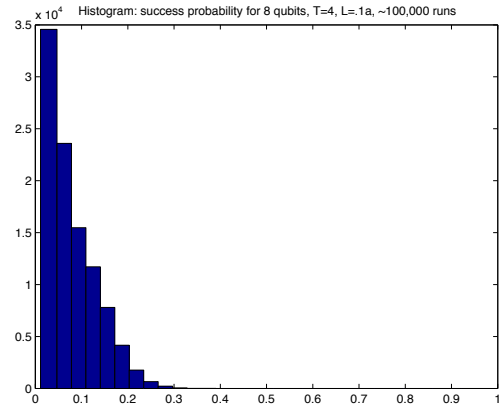


Figure 2. Results obtained via computation of Taylor series solutions of the 8 qubit master equation with the Lindblad operator $L = 0.1\hat{a}$ and $T = 4$. The probabilities of success outcomes obtained for 100,000 random Ising Hamiltonians H_f are sorted into 32 bins.

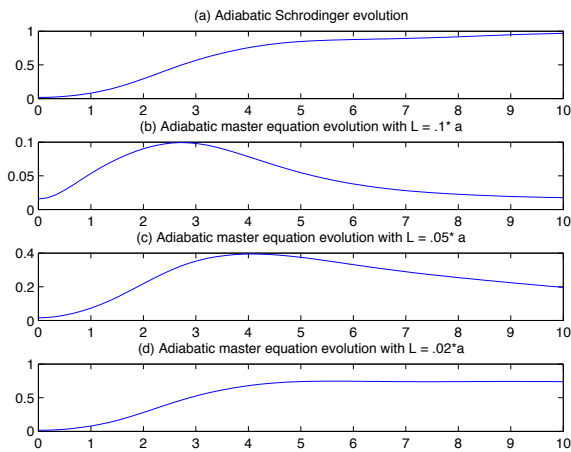


Figure 3. Dependence of the probability of success on T for solutions of the 8 qubit Schrödinger equation (a) and the master equations with the Lindblad operators $L = 0.1\hat{a}$ (b), $L = 0.05\hat{a}$ (c), and $L = 0.02\hat{a}$ (d). Note the inapplicability of the adiabatic theorem in (b) and (c), as well as convergence to the Schrödinger solution for diminishing norm of L .

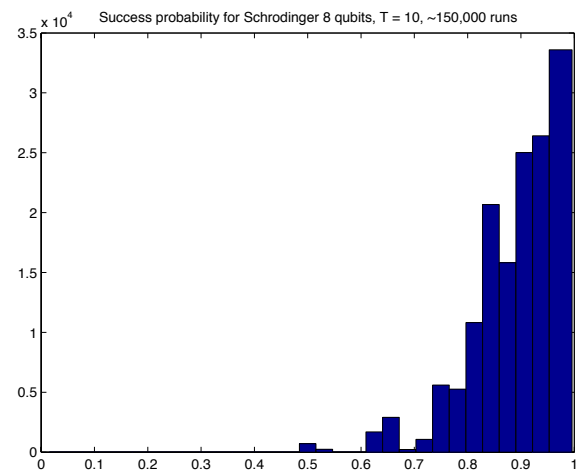


Figure 4. Schrödinger evolution as in Fig. 1 with 8 qubits but with $T = 10$. The histogram may be described as multi-modal.

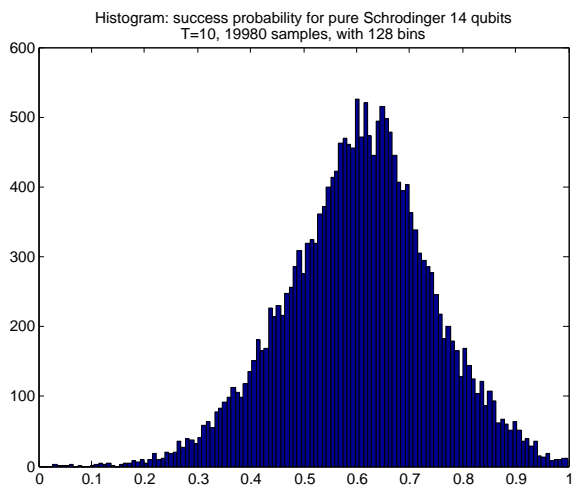


Figure 5. Schrödinger evolution with 14 qubits and $T = 10$. The computation of these results took about 12 minutes using 90 cores.

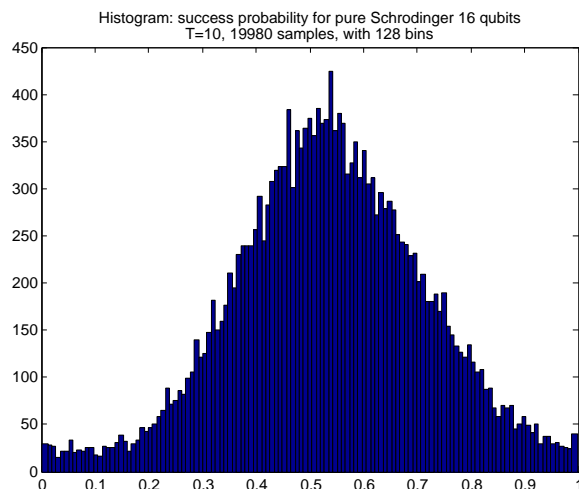


Figure 6. Schrödinger evolution with 16 qubits and $T = 10$. Computation time with 90 cores was 1 hour and 15 minutes approximately.

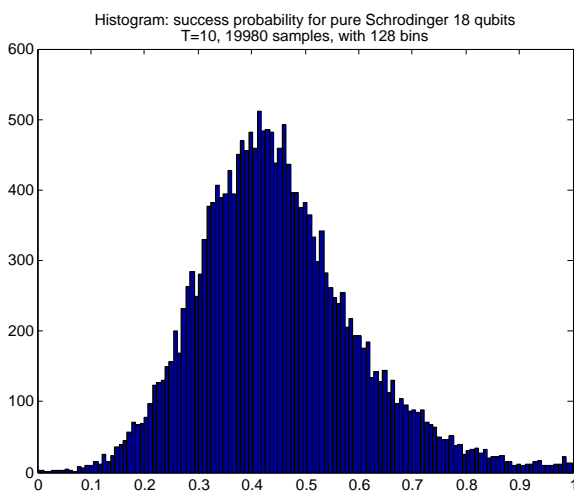


Figure 7. Schrödinger evolution with 18 qubits and $T = 10$. Computation time was approximately 5 hours and 20 minutes with 90 cores.

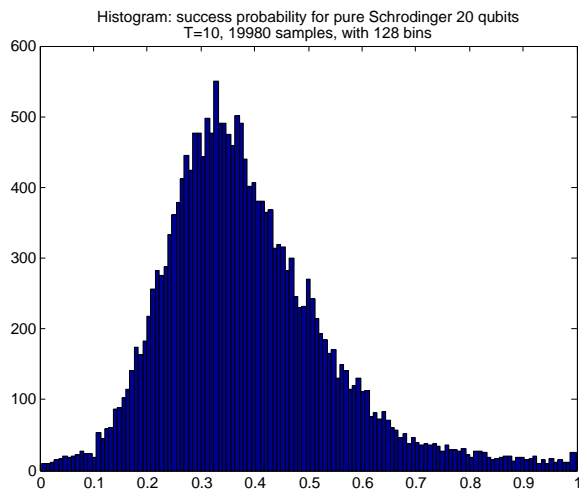


Figure 8. Schrödinger evolution with 20 qubits and $T = 10$. Using 180 cores, these results were computed in 14 hours.

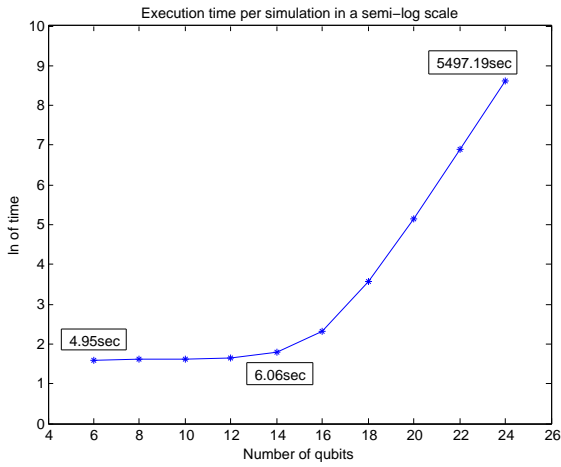


Figure 9. Execution time (wall time) of the algorithm as a function of the number of qubits. Our current implementation running on MATLAB implies a system time larger than the time required by the algorithm when the number of qubits is too small.

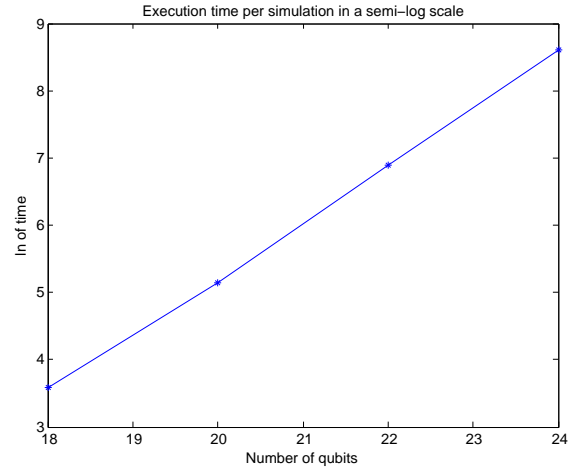


Figure 10. Execution time of the algorithm as a function of the number of qubits. A clear exponential behaviour is observed for larger number of qubits. The exponential function that fits the last three points ($x=20, 22$ and 24) is given by: $y = ce^{mx}$ where $c \simeq 6\mu s$, and $m = 0.86$.

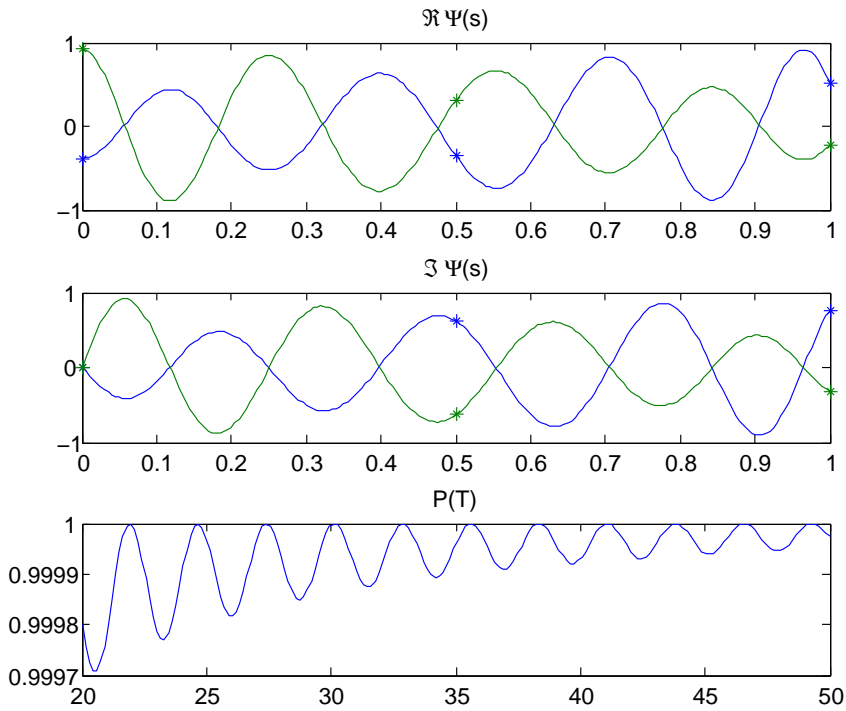


Figure 11. The top two figures show a numerical solution of (2) with $T/\hbar = 20$ and the Landau-Zener Hamiltonian $H(s) = (1 - 2s)\sigma_z + \Delta\sigma_x$ obtained via condensing the Taylor series to 200 small steps. The star marked points are obtained via computation in just two instalments. The first (resp. second) figure shows the real (resp. imaginary) parts of the two vector components. The bottom graph shows oscillations of the probability of success $P(T)$ as T/\hbar varies between 20 and 50.

Appendix II: Source codes

```

function P=Main(runs,N,T,nint,epsilon)

dim=2^N;
psi_0 = (1/sqrt(dim))*ones(dim/2,1);
H_i=Initial_Hamiltonian(N-1);
step=1/nint;
for n=1:runs
    H_f=Random_Ising(N);
    ind=(H_f == min(H_f));
    psi_in=psi_0;
    for k = 1:nint
        psi_in = Taylor_Installements(H_i,H_f,T,psi_in,step,tol,k);
    end
    P[n] = 2*norm(psi_in(ind))^2;
end

```

Function Main: Function that is being executed on each of the assigned cores on the cluster. Each core will obtain $n = \text{runs}$ samples. Observe how the Taylor series is calculated for n_{int} segments in the s -interval. N is the number of qubits, T the final time and tol the tolerance passed to the function `Taylor_Installements` below.

```

function H=Random_Ising(d)

dim=2^d;
vals=zeros(1,dim/2);
for k=1:d-1
    temp=zeros(1,dim/2^k);
    for r=k+1:d
        mac=[ones(1,dim/2^r) -ones(1,dim/2^r)];
        for p=1:r-(k+1)
            mac=[mac mac];
        end
        signo=2*randi(2,1)-3;
        temp=temp+signo*mac;
    end
    for p=1:k-1
        temp=[temp fliplr(temp)];
    end
    vals=vals+temp;
end
H=sparse(vals');

```

Function `Random_Ising`: Using a fractal-like iteration, creates a sparse vector H of dimension 2^d that represents the diagonal of an Ising Hamiltonian. Notice that this vector is palindrome, thus, only the first 2^{d-1} values are returned.


```

function H=Initial_Hamiltonian(d)

dim=2^d;
rows=[1 2];
cols=[2 1];
for k=1:d-1
    rows=[rows rows+2^k];
    cols=[cols cols+2^k];
    p=1:2^k;
    rows=[rows p p+2^k];
    cols=[cols p+2^k p];
end
vals=-ones(1,size(rows,2));
H = sparse(rows, cols, vals, dim, dim);

```

Function `Initial_Hamiltonian`: Using a fractal-like iteration, creates a sparse matrix H (the initial Hamiltonian) of dimension 2^d . Notice that for N qubits we use `Initial_Hamiltonian(N-1)` due to the symmetry in H_f .

```

function psi=Taylor_Installements(H_i,H_f_diag,T,psi_in,step,tol,k)

%***** Prepares Auxiliary Operators A and B of Eq. (6) for the recurrence
c = -1i * T;
d = (k - 1) * step;
i_by_psi_in = H_i * psi_in - flipud(psi_in);
f_by_psi_in = H_f_diag .* psi_in;
psi_n_min_2 = psi_in;
psi_n_min_1 = c * ((1-d) * i_by_psi_in + d * f_by_psi_in);
psi = psi_n_min_1 * step + psi_n_min_2;
nrm_cor = 1;
n = 1;
i_by_min_2 = i_by_psi_in;
f_by_min_2 = f_by_psi_in;

%***** Implements the recurrence as in Eq. (8)
while (nrm_cor > tol)
    n = n+1;
    i_by_min_1 = H_i * psi_n_min_1 - flipud(psi_n_min_1);
    f_by_min_1 = H_f_diag .* psi_n_min_1;
    psi_n = (c/n)*((1-d) * i_by_min_1 + d * f_by_min_1 + f_by_min_2 - i_by_min_2);
    cor = psi_n*step^n;
    nrm_cor = norm(cor);
    psi = psi + cor;
    psi_n_min_2 = psi_n_min_1;
    psi_n_min_1 = psi_n;
    i_by_min_2 = i_by_min_1;
    f_by_min_2 = f_by_min_1;
end

```

Function `Taylor_Installements`: Favouing vector-vector calculations, this function computes the Taylor series for the k -th segment of the s -interval. Notice that the calculation is stopped when the contribution at the next step is smaller than the tolerance `tol`.