

Constant geometric speed schedule for adiabatic state preparation

Mancheon Han^{1,*}, Hyowon Park^{2,3}, and Sangkook Choi^{1,†}

¹*School of Computational Sciences, Korea Institute for Advanced Study (KIAS), Seoul 02455, South Korea*

²*Materials Science Division, Argonne National Laboratory, Argonne, Illinois 60439, USA*

³*Department of Physics, University of Illinois at Chicago, Chicago, Illinois 60607, USA*



(Received 29 October 2025; revised 3 February 2026; accepted 9 May 2026; published 1 June 2026)

The efficiency of adiabatic quantum evolution is governed by the evolution time T , which typically scales as $\mathcal{O}(\Delta^{-2})$ with the minimum energy gap Δ . However, the rigorous lower bound is $\mathcal{O}(L\Delta^{-1})$, where L is the adiabatic path length. Although L is formally upper bounded by $\mathcal{O}(\Delta^{-1})$, such a bound is often too loose in practice, and L can be bounded independently of Δ . This indicates the potential for a quadratic speedup through adiabatic schedule construction. Here, we introduce the constant geometric speed (CGS) schedule, which traverses the adiabatic path at a uniform rate. We show that this approach reduces the scaling of the evolution time by a factor of Δ^{-1} , provided L remains bounded independently of Δ . We propose a segmented CGS protocol where path segment lengths are computed from eigenstate overlaps on the fly, reducing the prior spectral-knowledge requirement from the full gap function $\Delta(s)$ to just a global lower bound on the energy gap. Numerical tests on adiabatic unstructured search, N_2 , and a [2Fe-2S] cluster demonstrate the optimal Δ^{-1} scaling, confirming a quadratic speedup over the standard linear schedule.

DOI: 10.1103/ygs3-xgbl

I. INTRODUCTION

The adiabatic theorem [1–3] states that a quantum system remains in its instantaneous eigenstate as long as the state of interest is separated by an energy gap from other eigenstates and the Hamiltonian changes sufficiently slowly. This principle underlies both analog adiabatic quantum computation [4–6] and its digital counterpart, adiabatic state preparation (ASP) [7–9], which implements adiabatic evolution through quantum circuits [10] and is widely used as a state-preparation subroutine for algorithms such as quantum phase estimation [11,12]. In both cases, the key resource is the *adiabatic evolution time* T , defined here as the minimum duration required for the evolved state $|\psi(T)\rangle$ to achieve a target fidelity $\mathcal{F} = |\langle\psi(T)|\Phi_f\rangle|^2$ with the desired eigenstate $|\Phi_f\rangle$.

Among various parameters that affect the required adiabatic evolution time T , the minimal energy gap Δ plays a major role. Both lower and upper bounds for T in terms of Δ are established [6,13]. The adiabatic evolution starts with the eigenstate $|\Phi_i\rangle$ of the easily solvable Hamiltonian H_i , and aims to obtain the eigenstate $|\Phi_f\rangle$ of H_f , the target Hamiltonian. The rigorous adiabatic theorem [3] applies to any variation of the Hamiltonian between H_i and H_f as long as $H(t)$, the Hamiltonian as a function of time t , is a function

of the renormalized time $\tau = t/T$ that starts with H_i and ends at H_f . Here, we narrow down our discussion to the following form:

$$H(s(\tau)) = H_i + (H_f - H_i)s(\tau). \quad (1)$$

The parameter s is a monotonically increasing function of the normalized time τ , referred to as the schedule. The schedule $s(\tau)$ maps the normalized time $\tau \in [0, 1]$ to the interpolation parameter $s \in [0, 1]$, satisfying $s(0) = 0$ and $s(1) = 1$.

For the given adiabatic path $\{|\Phi(s)\rangle\}_{s \in [0,1]}$ of instantaneous eigenstates $|\Phi(s)\rangle$ of $H(s)$, the lower bound for T is given by [14]

$$T > \mathcal{O}(L/\Delta), \quad (2)$$

where $L = \int_0^1 \|\partial_s |\Phi(s)\rangle\| ds$ is the adiabatic path length. As L can often be bounded independently of Δ [14], the required adiabatic evolution time T scales at least as $\mathcal{O}(\Delta^{-1})$. On the other hand, a sufficient condition for adiabaticity [3,6] yields an upper bound

$$T < \mathcal{O}\left(\frac{\|\dot{H}\|}{\Delta^2}\right) + \mathcal{O}\left(\frac{\|\ddot{H}\|}{\Delta^2}\right) + \mathcal{O}\left(\frac{\|\ddot{H}\|^2}{\Delta^3}\right), \quad (3)$$

where $\dot{}$ denotes a derivative with respect to τ . This condition implies that T scales at most as $\mathcal{O}(\Delta^{-2})$ or even $\mathcal{O}(\Delta^{-3})$. These different scaling behaviors between upper and lower bounds highlight the potential role of scheduling, i.e., the choice of $s(\tau)$. For example, in the adiabatic Grover search, a linear schedule gives $T = \mathcal{O}(\Delta^{-2})$, whereas an optimized schedule using *a priori* gap information achieves the optimal $\mathcal{O}(\Delta^{-1})$ scaling [15]. Since then, a number of optimal scheduling strategies have been proposed [16–24]. However,

*Contact author: mchan@kias.re.kr

†Contact author: sangkookchoi@kias.re.kr

they require spectral information or lack general performance guarantees, which limits their practical applicability.

In this work, we introduce a geometric framework for constructing schedules that improve the gap scaling. Motivated by the geometric origin of the optimal $\mathcal{O}(\Delta^{-1})$ scaling, we propose the *constant geometric speed schedule*, which traverses the adiabatic path at a uniform rate and improves the scaling of the upper bound of T in $1/\Delta$ by one order, provided that L remains bounded independently of Δ . We then propose a practical classical-quantum hybrid algorithm to implement this schedule using eigenstate overlaps computed along the evolution. We demonstrate the method on the adiabatic Grover algorithm, N_2 molecule, and a [2Fe-2S] cluster, showing that the constant geometric speed schedule achieves the optimal $\mathcal{O}(\Delta^{-1})$ scaling, a quadratic speedup over $\mathcal{O}(\Delta^{-2})$ scaling of the linear schedule. In contrast to earlier approaches, our method reduces the prior spectral-knowledge requirement from the full gap function $\Delta(s)$ to just a global lower bound on the energy gap, and provably improves the gap dependence of the adiabatic evolution.

II. NOTATION

Before presenting the methods, we briefly summarize the notation. The Hamiltonian along the adiabatic path is denoted by $H(s)$, with H_i and H_f the end points and $|\Phi_i\rangle$, $|\Phi_f\rangle$ their respective eigenstates. The state at physical time t , $|\psi(t)\rangle$, evolves from the initial state $|\Phi_i\rangle$ according to the Schrödinger equation with $H(s(\tau))$. When no confusion arises, we write $H(\tau) \equiv H(s(\tau))$ and $|\Phi(\tau)\rangle$ for the instantaneous eigenstate $|\Phi(s(\tau))\rangle$.

III. ADIABATIC EVOLUTION ERROR

With $\mathcal{F} = |\langle\psi(T)|\Phi_f\rangle|^2$, the adiabatic evolution error satisfies [3]

$$1 - \mathcal{F} \leq \left(\frac{\mathcal{C}[s]}{T} \right)^2, \quad (4)$$

where the functional $\mathcal{C}[s]$ depends on the schedule $s(\tau)$ and is given by

$$\begin{aligned} \mathcal{C}[s] = & \sum_{\tau \in [0,1]} \frac{\|\dot{P}\|}{\Delta(\tau)} + \int_0^1 \frac{\|Q\ddot{P}P\|}{\Delta(\tau)} d\tau \\ & + \int_0^1 \frac{\|\dot{P}\|^2}{\Delta(\tau)} d\tau + \int_0^1 2 \frac{\|\dot{H}\| \|\dot{P}\|}{\Delta^2(\tau)} d\tau, \end{aligned} \quad (5)$$

with $P(\tau) = |\Phi(\tau)\rangle\langle\Phi(\tau)|$ and $Q(\tau) = \mathbb{I} - P(\tau)$. For clarity, the explicit τ dependence has been omitted. The evolution time T required to reach a desired fidelity is therefore proportional to $\mathcal{C}[s]$. Without exploiting the geometric structure of the eigenstate path, the terms in Eq. (5) can be bounded using derivatives of the Hamiltonian and the energy gap [3] as $\|\dot{P}\| \leq \|\dot{H}\|/\Delta$ and $\|Q\ddot{P}P\| \leq \|\dot{H}\|/\Delta + 4\|\dot{H}\|^2/\Delta^2$. Substituting these estimates yields upper bounds in Eq. (3).

IV. CONSTANT GEOMETRIC SPEED SCHEDULE

The lower bound of the adiabatic evolution time is determined by geometric properties of the eigenstate path [14],

suggesting that the scaling with respect to the minimum gap can be improved by rewriting Eq. (5) in geometric form. The natural parametrization of the path is its Fubini-Study arc length [25] $l(s) = \int_0^s \|\partial_{s'}\Phi(s')\| ds'$. Accordingly, the instantaneous speed is $v(\tau) = dl/d\tau = \|\dot{\Phi}(\tau)\|$. Since $\|\dot{P}\| = v(\tau)$, the first term of $\mathcal{C}[s]$ directly reflects this speed. The second term is associated with the curvature κ of the adiabatic path, which can be defined as [26] $\kappa(\tau) = \|Q d^2\Phi/dl^2\|$. Using the curvature, we have $\|Q\ddot{P}P\| = \sqrt{\dot{v}^2 + \kappa^2 v^4}$ (proof in Sec. I A of the Supplemental Material [27]). Thus, $\mathcal{C}[s]$ in Eq. (5) can be written with geometric terms as

$$\begin{aligned} \mathcal{C}[s] = & \sum_{\tau \in [0,1]} \frac{v(\tau)}{\Delta(\tau)} + \int_0^1 \frac{\sqrt{\dot{v}^2(\tau) + \kappa^2(\tau)v^4(\tau)}}{\Delta(\tau)} d\tau \\ & + \int_0^1 \frac{v^2(\tau)}{\Delta(\tau)} d\tau + \int_0^1 2 \frac{\|\dot{H}(\tau)\|v(\tau)}{\Delta^2(\tau)} d\tau. \end{aligned} \quad (6)$$

For the constant geometric speed (CGS) schedule $s_c(\tau)$, the speed becomes uniform, $v(\tau) = L$ with $L = \int_0^1 v(\tau) d\tau$, and the acceleration term $\dot{v}$ vanishes. Substituting into Eq. (6) yields

$$\begin{aligned} \mathcal{C}[s_c] = & L \sum_{\tau \in [0,1]} \frac{1}{\Delta(\tau)} + L^2 \int_0^1 \frac{\kappa(\tau) + 1}{\Delta(\tau)} d\tau \\ & + 2L \int_0^1 \frac{\|\dot{H}(\tau)\|}{\Delta^2(\tau)} d\tau. \end{aligned} \quad (7)$$

This leads to the following scaling for the total time T :

$$T = \mathcal{O}\left(\frac{L}{\Delta}\right) + \mathcal{O}\left(\frac{L(K+L)}{\Delta}\right) + \mathcal{O}\left(\frac{L\|\dot{H}\|}{\Delta^2}\right), \quad (8)$$

where $K = \int_0^L \kappa(l) dl$ is the total curvature of the adiabatic path. Comparing this with Eq. (3), every term in the bound improves by one order in Δ . Here, the explicit dependence on the gap is replaced by the geometric quantities L and K . If these geometric quantities exhibit worst-case scaling (e.g., $L \propto \Delta^{-1}$), the advantage of the CGS schedule vanishes. However, in many cases of interest, L [14] and K can be bounded independently of Δ [see Figs. 3(c) and 4(d)]. In this regime, the CGS schedule achieves a one-order improvement in gap scaling. Consequently, the constant geometric speed schedule can yield the optimal $T = \mathcal{O}(\Delta^{-1})$ scaling in cases where generic schedules typically require $T = \mathcal{O}(\Delta^{-2})$, such as for Hamiltonians belonging to the Gevrey class [28].

The constant geometric speed condition ($\dot{v} = 0$) implies $ds/d\tau \propto (dl/ds)^{-1}$. Since the geometric speed is bounded by $dl/ds \leq \|H_f - H_i\|/\Delta$, this suggests a similarity to previous gap-based locally optimal schedules $ds/d\tau \propto \Delta^p$ [3,6,15]. This similarity is supported by our numerical experiments (Fig. S2 in the Supplemental Material [27]). However, despite this similarity in performance, our method offers a distinct practical advantage. Unlike the gap function $\Delta(s)$, which requires knowledge of excited states, the geometric speed is determined solely by the ground state. Consequently, the CGS schedule can be computed on the fly during the adiabatic evolution, as the evolving state provides a good approximation of the true ground state. The only parameter required is a global lower bound of the energy gap Δ to filter out excited-state

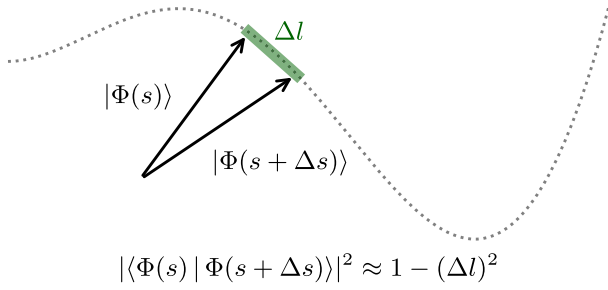


FIG. 1. Schematic construction of the CGS schedule. The segment length Δl is obtained from the overlap $|\langle \Phi(s) | \Phi(s + \Delta s) \rangle|^2$, and then Δt is adjusted so that $\Delta l / \Delta t$ remains constant.

components. This parameter is far more tractable than characterizing $\Delta(s)$ for all s , and can even be estimated adaptively. A related geometric formulation by Chen [24] likewise casts the optimal protocol as constant velocity traversal, but with respect to a *dynamical* quantum geometric tensor, in contrast to the standard Fubini-Study metric used in our work. This choice of metric allows our method to compute the adiabatic path length and the corresponding schedule directly through eigenstate overlaps.

We now describe how to construct such a schedule in practice based on the length of adiabatic path segments,

$$\Delta l = \int_s^{s+\Delta s} \left\| \frac{d|\Phi(s')\rangle}{ds'} \right\| ds'. \quad (9)$$

The equation suggests that we can traverse the adiabatic path at a uniform rate by keeping the ratio $\Delta l / \Delta t$ constant. The important thing to remark is that the segment length Δl can be obtained from the overlap between nearby eigenstates, from $|\langle \Phi(s) | \Phi(s + \Delta s) \rangle|^2 = 1 - (\Delta l)^2$ up to the leading order (see Fig. 1). Then, the following theorem provides a discretized approximation to the constant geometric speed schedule (proof in Sec. I B of the Supplemental Material [27]).

Theorem 1. Suppose a monotonic sequence $s_0 = 0 < s_1 < \dots < s_m = 1$ is given, and that the eigenstate overlaps

$$|\langle \Phi(s_j) | \Phi(s_{j+1}) \rangle|^2 = 1 - (\Delta l_{j+1})^2$$

are known for each adjacent pair (s_j, s_{j+1}) . Then, any schedule $\hat{s}(\tau)$ satisfying $\hat{s}(\tau_j) = s_j$ and

$$\tau_{j+1} = \tau_j + \frac{\Delta l_{j+1}}{\sum_{i=1}^m \Delta l_i}$$

converges to the constant geometric speed schedule s_c in the limit $\max_j (s_{j+1} - s_j) \rightarrow 0$.

Based on Theorem 1, we can construct a segmented constant geometric speed schedule, $\hat{s}_c(\tau)$, by interpolating the discrete points $\{(\tau_j, s_j)\}$. This schedule serves as a practical, discretized realization of a true constant geometric speed schedule. The core challenge in this approach lies in the computation of the path segment lengths Δl_j , which requires calculating the eigenstate overlaps $|\langle \Phi(s_j) | \Phi(s_{j+1}) \rangle|^2$. The practical algorithm for this task is given in the Appendix.

V. APPLICATIONS

We apply our method to a variety of quantum systems, ranging from quantum search problems to quantum chemistry. In the following analysis, we account for the overlap calculation overhead, T_{ov} , which we estimated as $4\Delta^{-1}$ (details of this estimate and other computational parameters are provided in the Appendix). For chemical systems, the B3LYP exchange-correlation functional [29,30] is used in a density functional theory (DFT) calculation.

A. Adiabatic Grover search

The Grover search problem [31] is a canonical example for adiabatic scheduling. Its adiabatic formulation [15] evolves the initial state $|\Phi_i\rangle = |+\rangle^{\otimes n}$ toward the marked computational basis state $|m\rangle$ using the Hamiltonians $H_i = \mathbb{I} - |\Phi_i\rangle\langle\Phi_i|$ and $H_f = \mathbb{I} - |m\rangle\langle m|$, interpolated as in Eq. (1). While a linear schedule $s(\tau) = \tau$ yields $T = \mathcal{O}(N)$, offering no advantage over classical search, the optimal schedule achieves a quadratic speedup. However, realizing this optimal schedule traditionally relies on the condition $\dot{s} \propto \Delta^2(s)$ [15], which necessitates explicit prior knowledge of $\Delta(s)$. Crucially, we prove that this optimal schedule is, in fact, identical to the CGS schedule (proof in Sec. I C of the Supplemental Material [27]). Consequently, our method naturally reproduces the optimal Grover schedule by computing local geometric information on the fly, without requiring *a priori* knowledge of $\Delta(s)$.

Figures 2(a)–2(c) show the numerical results for the adiabatic Grover problem with $N = 2^{14} = 16384$. Panel (a) displays the energy spectrum, which exhibits its minimum gap at $s = 0.5$. Consistent with this feature, our algorithm generates a schedule $\hat{s}_c(\tau)$ that slows down near the gap minimum, as shown in panel (b). The computed schedule points (red) lie almost exactly on the theoretical constant geometric speed schedule $s_c(\tau)$, with deviations arising only from interpolation. Panel (c) quantifies the resulting speedup: Our schedule reaches 75% fidelity approximately 92 times faster than the linear schedule ($T = 3.19 \times 10^2$ versus $T = 2.94 \times 10^4$).

Finally, Fig. 2(d) confirms the asymptotic scaling of the evolution time with the minimum gap on the adiabatic path Δ . For the adiabatic Grover search, $\Delta = 1/\sqrt{N}$ [15], so the plot also directly connects evolution time with the number of items N in the database. We plot the adiabatic evolution time T and the total time $T_{\text{tot}} = T + T_{\text{ov}}$. The numerical fit shows that the linear schedule's evolution time scales as $T \propto \Delta^{-2} = \mathcal{O}(N)$, while our constant geometric speed schedule achieves the optimal scaling of $T \propto \Delta^{-1} = \mathcal{O}(\sqrt{N})$. Since the overlap calculation overhead T_{ov} also scales as Δ^{-1} , the total time T_{tot} maintains this optimal scaling, with a larger prefactor.

B. N_2

The first system we consider is the nitrogen molecule, described using an STO-3G basis set [32] and restricted to the singlet subspace. As the initial Hamiltonian H_i , we use the Kohn-Sham Hamiltonian from a converged DFT calculation. The system then evolves toward the final Hamiltonian H_f , which is the full electronic Hamiltonian within this basis set. Figure 3(a) shows the two lowest-lying energies as a function

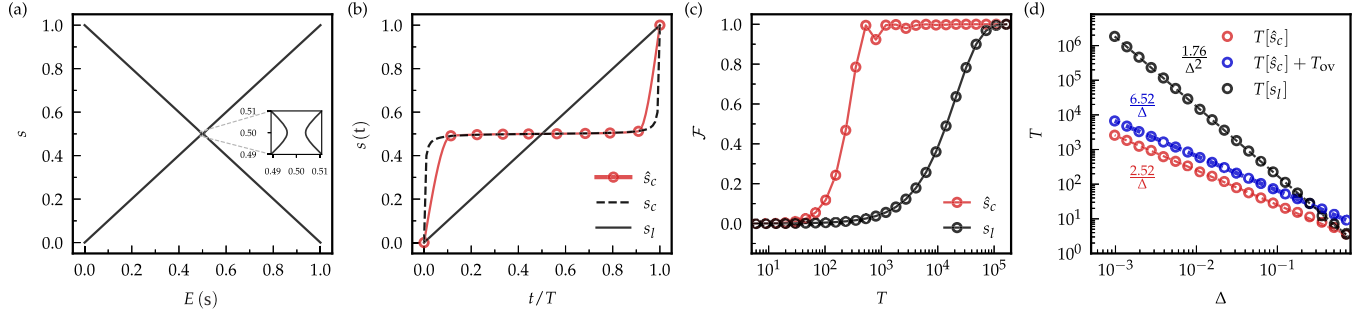


FIG. 2. Application of the constant geometric speed schedule to the adiabatic Grover search. Panels (a)–(c) show results for $N = 2^{14}$. (a) Energy spectrum of $H(s)$. (b) Segmented CGS schedule $\hat{s}_c$ compared with the exact optimal schedule s_c and the linear schedule s_l ; circles denote computed points. (c) Fidelity $\mathcal{F}$ vs evolution time T for $\hat{s}_c$ and s_l . (d) Evolution time required to reach $\mathcal{F} = 0.75$ as a function of the minimum gap Δ ; circles show numerical data and dashed lines indicate fitted scalings.

of bond length R , obtained from the DFT calculation corresponding to H_i and from a full configuration interaction (FCI) calculation of H_f . The inset reveals that the energy gap of the initial Hamiltonian decreases rapidly with increasing R , causing the overall minimum gap Δ of the adiabatic path to occur near $s = 0$ for large R .

Our CGS schedule $\hat{s}_c$ allocates a significantly larger fraction of the evolution time to this region, resulting in a substantial reduction in the required evolution time, as shown for $R = 3.5$ Å in Fig. 3(b) [see Fig. S1(a) of the Supplemental Material [27] for the shape of $\hat{s}_c$]. To be quantitative, let T denote the time required to reach a target fidelity of 0.75. We find that our schedule yields a speedup of $T[s_l]/T[\hat{s}_c] \approx 52.6$ over the linear schedule s_l . To investigate the origin of this speedup, in Fig. 3(c), we plot the adiabatic path length L and total curvature K as a function of the minimum gap Δ , obtained by varying the bond length R . We observe that both L and K remain bounded independently of Δ , unlike the theoretical worst-case scaling. This behavior accounts for the performance of CGS and leads to the reduced gap scaling from Δ^{-2} to Δ^{-1} , as confirmed in Fig. 3(d).

C. [2Fe-2S]

We demonstrate our method on a [2Fe-2S] cluster, a common bioinorganic motif and a representative strongly correlated system in quantum chemistry [33–36]. Following Lee

et al. [37], we test a wide range of initial Slater determinants and reproduce their observation that ASP performance under a linear schedule is highly unpredictable: The required evolution time varies over eight orders of magnitude depending on the initial state and often exceeds the estimated cost of quantum phase estimation (T_{QPE}) [37]. This extreme sensitivity arises from the $T \propto \Delta^{-2}$ scaling, which strongly amplifies variations in the minimum gap Δ .

Our constant geometric speed schedule mitigates this issue by achieving the optimal $T \propto \Delta^{-1}$ scaling. To quantitatively compare the two approaches across many initial states, we use the heuristic estimate [37]

$$T_{\text{ASP}}^{\text{est}} = \max_{\tau \in [0,1]} \frac{|\langle \Phi(\tau) | \dot{H}(\tau) | \mathcal{E}(\tau) \rangle|}{\Delta^2(\tau)}, \quad (10)$$

where $|\mathcal{E}(\tau)\rangle$ is the first excited state at τ .

The results in Fig. 4(b) confirm that the optimal gap scaling leads to improved reliability. While $T_{\text{ASP}}^{\text{est}}$ for the linear schedule (black) shows the previously reported large variation, the CGS schedule (red) exhibits much smaller variance and consistently stays below T_{QPE} (dotted line). By achieving optimal scaling, our method thus makes ASP robust and less unpredictable.

Figure 4(c) presents the actual adiabatic evolution results for a representative case, where the initial state corresponds to the DFT ground state [indicated by crosses in Fig. 4(b)].

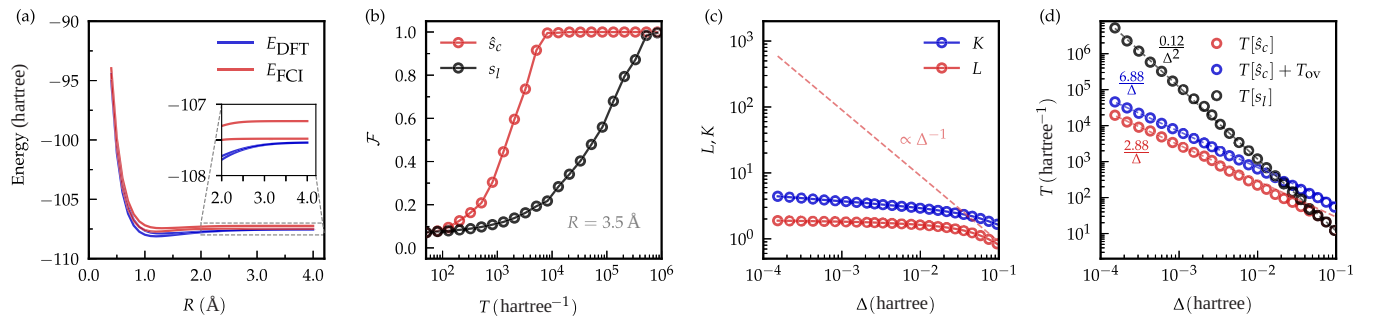


FIG. 3. Application of the CGS schedule to the nitrogen molecule. (a) Low-lying energy spectra from DFT (blue) and FCI (red) calculations. (b) Fidelity $\mathcal{F}$ as a function of the evolution time T at bond length $R = 3.5$ Å. (c) The adiabatic path length L and the total curvature K as a function of Δ . Dotted red line corresponds to worst-case scaling of L . (d) Evolution time needed to reach $\mathcal{F} = 0.75$ vs the minimum gap Δ . The constant geometric speed schedule (red) achieves the optimal Δ^{-1} scaling, while the linear schedule (black) follows Δ^{-2} . The total time including overlap overhead (blue) also follows the optimal scaling.

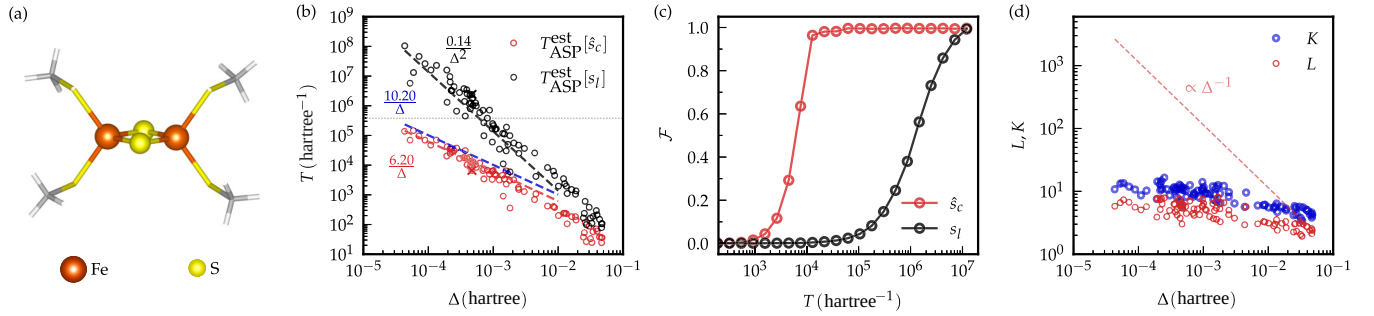


FIG. 4. Application of the CGS schedule to the [2Fe-2S] cluster. (a) Molecular structure. (b) Estimated adiabatic evolution time $T_{\text{ASP}}^{\text{est}}$ vs the minimum energy gap Δ , showing the optimal Δ^{-1} scaling for $\hat{s}_c$ (red) compared with the Δ^{-2} scaling of s_l (black). (c) Fidelity $\mathcal{F}$ as a function of evolution time T for the initial state chosen as the DFT ground state, corresponding to the crosses in panel (b). Plot shows a speedup of more than two orders of magnitude with $\hat{s}_c$. (d) The adiabatic path length L and the total curvature K as a function of Δ . Dotted red line corresponds to worst-case scaling of L .

The estimates in panel (b) suggest a speedup of more than two orders of magnitude with the CGS schedule, and the simulation in panel (c) confirms this expectation [see Fig. S1(b) of the Supplemental Material [27] for the shape of $\hat{s}_c$]. Using the CGS schedule, the system achieves a fidelity of 75% with $T = 9.48 \times 10^3$, which is approximately 289 times faster than the linear schedule ($T = 2.74 \times 10^6$). Consistent with the findings in the N_2 molecule, the observed performance gain and the reduced gap scaling are directly attributed to the geometric properties shown in Fig. 4(d), where both the adiabatic path length L and the total curvature K remain bounded independently of the minimum gap Δ .

VI. CONCLUSION

We have introduced the constant geometric speed (CGS) schedule as a general strategy to improve the gap scaling of adiabatic state preparation. By enforcing uniform evolution along the eigenstate path, the method reduces the gap dependence of the run-time by one order compared with generic schedules, provided that the adiabatic path length L remains bounded independently of Δ . Furthermore, its overlap-guided construction makes the approach practical without requiring *a priori* spectral knowledge. Our numerical results on the adiabatic Grover search, the nitrogen molecule, and the [2Fe-2S] cluster demonstrate a clear quadratic speedup, improving the adiabatic scaling from $\mathcal{O}(\Delta^{-2})$ to $\mathcal{O}(\Delta^{-1})$ and substantially enhancing the reliability of state preparation. Our numerical experiments indicate that the gap-independent boundedness of geometric quantities L and K is a typical feature in practice, supporting the broad applicability of the proposed method. While this empirical observation emerges consistently across our tested systems, a rigorous characterization of the Hamiltonian classes exhibiting such gap-independent geometric boundedness, together with a formal proof of schedule optimality, is left for future work. Overall, our findings indicate that exploiting the geometry of the adiabatic path provides an effective strategy for improving quantum simulation.

ACKNOWLEDGMENTS

We used resources of the Center for Advanced Computation at Korea Institute for Advanced Study and the National

Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231. S.C. was supported by a KIAS Individual Grant (No. CG090601) at Korea Institute for Advanced Study and by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea Government (MSIT) (No. 2022-0-01026). M.H. was supported by a KIAS Individual Grant (No. CG091302) at Korea Institute for Advanced Study and by Quantum Simulator Development Project for Materials Innovation through the National Research Foundation of Korea (NRF) funded by the Korean government [Ministry of Science and ICT(MSIT)] (No. NRF-2023M3K5A1094813).

DATA AVAILABILITY

The numerical data, figures, and the code used to reproduce them are openly available at Ref. [38].

APPENDIX: PRACTICAL ALGORITHM AND COMPLEXITY ANALYSIS

Here, we discuss the practical quantum algorithm to construct the segmented CGS $\hat{s}_c(\tau)$ based on Theorem 1. The key step is to compute the eigenstate overlaps $|\langle \Phi(s_j) | \Phi(s_{j+1}) \rangle|^2$. One approach to calculate this overlap is through the use of projection operators, $P(s) = |\Phi(s)\rangle \langle \Phi(s)|$. The squared overlap can be computed from the ratio of two expectation values:

$$|\langle \Phi(s_j) | \Phi(s_{j+1}) \rangle|^2 = \frac{p_j(s_{j+1})}{f_j}, \quad (\text{A1})$$

where

$$f_j = \|P(s_j)|\psi(t_j)\rangle\|^2, \quad (\text{A2})$$

$$p_j(s_{j+1}) = \|P(s_{j+1})P(s_j)|\psi(t_j)\rangle\|^2. \quad (\text{A3})$$

The quantity f_j serves as a fidelity estimate between the evolved state $|\psi(t_j)\rangle$ and the instantaneous eigenstate $|\Phi(s_j)\rangle$, and it remains non-negligible as long as the evolution time is sufficiently long. Consequently, the segment length is com-

ALGORITHM 1. Segmented CGS schedule construction.

Input: Target segment length Δl_t , time for the first step t_1 .
Output: Segmented CGS schedule $\hat{s}_c(\tau)$.
Initialize: $j \leftarrow 0, t_0 \leftarrow 0, s_0 \leftarrow 0$, reference length $\Delta l_r \leftarrow 0$.
Initialize: Schedule points $\mathcal{S} \leftarrow \{(t_0, s_0)\}$.
while $s_j < 1$ **do**
 Construct the schedule $\hat{s}(t)$ by interpolating points in $\mathcal{S}$.
 Prepare $|\psi(t_j)\rangle$ by evolving $|\Phi(s_0)\rangle$ for time t_j using $\hat{s}(t)$.
 Define $g(s') \leftarrow p_j(s')/f_j - (1 - \Delta l_t^2)$.
 Find $s_{j+1} > s_j$ that solves $g(s_{j+1}) = 0$.
 Compute $\Delta l_{j+1} \leftarrow \sqrt{1 - p_j(s_{j+1})/f_j}$.
 if $j = 0$ **then**
 Set reference length $\Delta l_r \leftarrow \Delta l_1$.
 Set next time point $t_{j+1} \leftarrow t_1$.
 else
 Set next time point $t_{j+1} \leftarrow t_j + t_1 \cdot (\Delta l_{j+1}/\Delta l_r)$.
 end if
 Add the new point (t_{j+1}, s_{j+1}) to $\mathcal{S}$.
 $j \leftarrow j + 1$.
end while
Set total evolution time $T \leftarrow t_j$.
Construct final schedule $\hat{s}(t)$ by interpolating all points in $\mathcal{S}$.
Return normalized schedule $\hat{s}_c(\tau) = \hat{s}(T\tau)$.

puted as

$$\Delta l_{j+1} = \sqrt{1 - \frac{p_j(s_{j+1})}{f_j}}. \quad (\text{A4})$$

This procedure provides a general framework for constructing the schedule, assuming one has a method to implement the required projection operators. The entire procedure is summarized in Algorithm 1.

At first glance, it might appear simpler to compute segment lengths at predetermined uniform grid points rather than using the root-finding procedure in Algorithm 1. However, as seen in the adiabatic Grover problem (Fig. 2), the energy gap $\Delta(s)$ often becomes small only in a very narrow region of s , so a uniform grid may fail to capture the point where long evolution time is required. In contrast, our root-finding procedure adaptively places grid points in proportion to the required time, ensuring that such regions are properly resolved.

A concrete implementation of the projection operators required in Algorithm 1 can be realized using the quantum Zeno Monte Carlo (QZMC) method [39]. The QZMC method approximates the projector with a Gaussian that can be represented as an integral of time evolution,

$$\begin{aligned} P(s) &\approx e^{-\frac{\beta^2}{2}(H(s)-E(s))^2} \\ &= \frac{1}{\sqrt{2\pi\beta^2}} \int_{-\infty}^{\infty} e^{-\frac{t^2}{2\beta^2}} e^{-i(H(s)-E(s))t} dt, \end{aligned} \quad (\text{A5})$$

where $E(s)$ is the energy eigenvalue of $H(s)$ corresponding to $|\Phi(s)\rangle$. This approach requires two additional parameters: β , which determines the width of the projection operator, and the number of Monte Carlo samples N_v to construct integral in Eq. (A5). The accuracy of this approximation and the associated resources are governed by these two parameters. Furthermore, the energy eigenvalue $E(s)$ is determined using

the method described in Sec. II of the Supplemental Material [27].

Next, we analyze the computational cost of the proposed algorithm in terms of circuit depth and iteration count. In this analysis, the target segment length Δl_t is treated as a constant parameter based on the following reasoning. Since the primary role of the constant geometric speed schedule is to distribute the evolution time in proportion to the geometric length, it is sufficient that Δl_t remains small compared to the total path length L to obtain a practical advantage. This reasoning is supported by our numerical experiments (Figs. 2–4), where a fixed value of $\Delta l_t = 0.2$ was employed and demonstrated consistent scaling improvements.

The *circuit depth* is determined by the total time evolution length, which consists of the adiabatic evolution time T and the additional time length T_{ov} required to implement the projection operator. The latter scales as $\mathcal{O}(\beta)$ [39], where β is the parameter in the Gaussian approximation of the projection operator in Eq. (A5). Therefore, the total time evolution length is

$$T_{\text{tot}} = T + \mathcal{O}(\beta). \quad (\text{A6})$$

Here, T is a adiabatic evolution time with the constant geometric speed schedule. For our algorithm to be reliable, the adiabatic evolution time T should be large enough to ensure $f_j > 1 - \eta$ for all j , with η taken not too large (e.g., $\eta < 0.5$).

To construct the projector that resolves the target eigenstate from nearby eigenstates, β should satisfy $\beta > \Delta_{\min}^{-1}$, where $\Delta_{\min}$ denotes the minimum energy gap in the adiabatic path, i.e., $\Delta_{\min} = \min_{\tau} \Delta(\tau)$. In our numerical experiments, setting $\beta = 2\Delta_{\min}^{-1}$ was sufficient. Therefore,

$$T_{\text{tot}} = T + \mathcal{O}(\Delta_{\min}^{-1}). \quad (\text{A7})$$

This shows that the additional cost from the projection operator, $\mathcal{O}(\Delta_{\min}^{-1})$, matches the optimal scaling [Eq. (2)] of adiabatic algorithms with respect to the minimum energy gap. Therefore, the projection does not worsen the overall scaling of the evolution time. Further rigorous analysis shows that the required β also depends on parameters such as the target segment length Δl_t and the precision ϵ of the estimate $p_j(s_{j+1})/f_j$, which are independent of the Hamiltonian, so is not discussed here. Such details are provided in Sec. II of the Supplemental Material [27].

The *iteration count* of the algorithm is determined by two factors: (1) the number of schedule segments m and (2) the number of Monte Carlo evaluations required for each segment.

The number of segments is determined by the overlap condition

$$|\langle \Phi(s_j) | \Phi(s_{j+1}) \rangle|^2 = 1 - \Delta l_t^2, \quad \Delta l_{j+1} = \Delta l_t. \quad (\text{A8})$$

Since the total path length L is approximately equal to $\sum_j \Delta l_j$, the number of segments is

$$m \approx \frac{L}{\Delta l_t}. \quad (\text{A9})$$

The primary computational task within each segment is the root-finding search for the next point s_{j+1} that satisfies Eq. (A8). This search is iterative, requiring multiple evalua-

tions of the overlap. We define $\bar{N}_{\text{root}}$ as the average number of these evaluations needed per segment. In addition to these $\bar{N}_{\text{root}}$ steps, a separate evaluation is required to compute the fidelity term f_j . Therefore, the total number of Monte Carlo evaluations per segment is approximately $\bar{N}_{\text{root}} + 1$. Each of these evaluations uses N_v Monte Carlo samples, and the standard deviation for this statistical estimation is bounded by $\sqrt{2N_v^{-1/2}(1-\eta)^{-1}}$ (see Sec. II C of the Supplemental Material [27] for derivation). To ensure the error in the overlap estimation is below a threshold ϵ , it suffices to set $N_v \geq 2\epsilon^{-2}(1-\eta)^{-2}$. Combining these factors, the total number of Monte Carlo samples required, $N_{\text{total}} = m \times (\bar{N}_{\text{root}} + 1) \times N_v$, is

$$N_{\text{total}} \approx (\bar{N}_{\text{root}} + 1) \frac{L}{\Delta_l} \frac{2}{\epsilon^2} \frac{1}{(1-\eta)^2}. \quad (\text{A10})$$

Having expressed the repetitions in terms of the root-finding steps, we now theoretically analyze the complexity of N_{root} . In our algorithm, determining the increment Δs requires solving $|\langle \Phi(s_j) | \Phi(s_j + \Delta s) \rangle|^2 = 1 - \Delta l_i^2$. Our implementation utilizes a bracketing strategy followed by a root-finding method (e.g., Brent's method): We start with a safe lower bound δs and iteratively double the step size ($\Delta s_{\text{trial}} \leftarrow 2\Delta s_{\text{trial}}$) until the interval $[0.5\Delta s_{\text{trial}}, \Delta s_{\text{trial}}]$ encloses the solution, after which the root is refined within this bracket. To guarantee that the initial search does not overlook the solution in regions with large geometric speed $v_{\text{max}} \propto \Delta^{-1}$, the lower bound δs must scale as $\delta s \lesssim \Delta l_i / v_{\text{max}} \propto \Delta$. Since Δs is at most 1, the maximum number of doubling steps required to cover the dynamic range scales as

$$N_{\text{bracket}} \leq \log_2 \left(\frac{1}{\delta s} \right) = \mathcal{O}(\log_2(\Delta^{-1})). \quad (\text{A11})$$

Upon completion of the bracketing phase, the root is confined to an interval of width $W = 0.5\Delta s_{\text{trial}}$. Since the true step size Δs falls within this interval ($0.5\Delta s_{\text{trial}} \leq \Delta s \leq \Delta s_{\text{trial}}$), the interval width W scales proportionally with Δs . Consequently, for a target relative precision ϵ_{rel} , the number of refinement steps is given by $\log_2(W/(\epsilon_{\text{rel}}\Delta s)) \approx \log_2(1/\epsilon_{\text{rel}})$, which is independent of the gap. Therefore, the total complexity of the root-finding procedure per segment becomes

$$N_{\text{root}} = \mathcal{O}(\log_2(\Delta^{-1})), \quad (\text{A12})$$

confirming that the root-finding overhead scales only logarithmically with the inverse gap.

Next, we analyze the gap scaling of the cumulative run-time, T_{run} , defined as the sum of the total time evolution lengths over all segments and repetitions. For the j th segment, the evolution time required for a single circuit instance is given by

$$T_{\text{tot},j} \approx \frac{j}{m} T + \mathcal{O}(\Delta^{-1}). \quad (\text{A13})$$

This circuit is repeated $(N_{\text{root},j} + 1)N_v$ times to estimate the overlap. Since the number of repetitions $(N_{\text{root},j} + 1)N_v$ scales at most logarithmically with Δ^{-1} as discussed previously,

the cumulative run-time $T_{\text{run}} = \sum_{j=0}^{m-1} T_{\text{tot},j}(N_{\text{root},j} + 1)N_v$ satisfies

$$T_{\text{run}} < \mathcal{O}(\log_2(\Delta^{-1})) \sum_{j=0}^{m-1} T_{\text{tot},j} \\ \lesssim \mathcal{O}(\log_2(\Delta^{-1})) \left(\frac{mT}{2} + m\mathcal{O}(\Delta^{-1}) \right). \quad (\text{A14})$$

Given that the number of segments scales as $m = \mathcal{O}(L)$, we obtain

$$T_{\text{run}} < \mathcal{O}(L \log_2(\Delta^{-1}))(T + \mathcal{O}(\Delta^{-1})). \quad (\text{A15})$$

If L exhibits worst-case scaling ($L \sim \Delta^{-1}$), the adiabatic evolution time T is lower bounded by $\mathcal{O}(\Delta^{-2})$, leading to a cumulative run-time of $T_{\text{run}} \sim \mathcal{O}(\Delta^{-3})$ (up to logarithmic corrections). This scaling is indeed less favorable than that of the standard linear schedule. However, in cases where L remains bounded independently of Δ , T_{run} scales linearly with T (up to a logarithmic factor). Consequently, if the CGS schedule achieves $T = \mathcal{O}(\Delta^{-1})$, the cumulative run-time also follows $T_{\text{run}} \sim \mathcal{O}(\Delta^{-1})$. This confirms that the quadratic speedup of our method persists even after considering the total cost.

In addition to the above theoretical analysis, we specify here the parameter choices used in our practical implementation of the algorithm for the numerical experiments in Figs. 2–4. We used $\beta = 2\Delta^{-1}$ and estimated $T_{\text{ov}} = 2\beta = 4\Delta^{-1}$ as a reasonable overhead for the projection operations, because at most two projection operations are required to compute each overlap. Moreover, we used $\Delta l_i = 0.2$ and $N_v = 10^4$, which yielded accurate schedules with moderate computational cost. Under these parameter settings, the average number of root-finding steps per segment was observed to be $\bar{N}_{\text{root}} < 20$.

Finally, we highlight several practical aspects of the algorithm. First, in many applications, the minimum gap Δ is not known beforehand. For such cases, a suitable value of β can be found empirically by starting with a small value and iteratively increasing it (e.g., along a geometric sequence $\beta_k = 2^k \beta_0$) until convergence of the resulting schedule is reached. Second, the initial segment time t_1 sets the overall timescale for the evolution via the relation $T \approx t_1 \times (L/\Delta l_i)$. A suitable value for t_1 can be estimated from the application of Eq. (5) through a perturbative expansion $|\Phi(s)|$ near $s = 0$ or can be adjusted iteratively until the fidelity f_1 exceeds a desired threshold. Furthermore, the root-finding step in Algorithm 1 need not be highly precise. Theorem 1 does not require the segment lengths Δl_j to be uniform; it is sufficient to identify a point s_{j+1} that yields an overlap close to the target value, so only a few root-finding steps are necessary. Finally, for constructing a continuous schedule from the discrete points, we employed monotonic cubic spline interpolation [40], as implemented in SCIPY [41]. Our numerical results in Figs. 2–4 show that while this interpolation method can introduce small variations in the final fidelity, it does not affect the scaling of the adiabatic evolution time with respect to the minimum gap.

[1] M. Born and V. Fock, Beweis des Adiabatenatzes, *Z. Phys.* **51**, 165 (1928).

[2] T. Kato, On the adiabatic theorem of quantum mechanics, *J. Phys. Soc. Jpn.* **5**, 435 (1950).

- [3] S. Jansen, M.-B. Ruskai, and R. Seiler, Bounds for the adiabatic approximation with applications to quantum computation, *J. Math. Phys.* **48**, 102111 (2007).
- [4] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Quantum computation by adiabatic evolution, [arXiv:quant-ph/0001106](https://arxiv.org/abs/quant-ph/0001106).
- [5] I. M. Georgescu, S. Ashhab, and F. Nori, Quantum simulation, *Rev. Mod. Phys.* **86**, 153 (2014).
- [6] T. Albash and D. A. Lidar, Adiabatic quantum computation, *Rev. Mod. Phys.* **90**, 015002 (2018).
- [7] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, Simulated quantum computation of molecular energies, *Science* **309**, 1704 (2005).
- [8] L. Veis and J. Pittner, Adiabatic state preparation study of methylene, *J. Chem. Phys.* **140**, 214111 (2014).
- [9] C. Mc Keever and M. Lubasch, Towards adiabatic quantum computing using compressed quantum circuits, *PRX Quantum* **5**, 020362 (2024).
- [10] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, UK, 2010).
- [11] A. Y. Kitaev, Quantum measurements and the Abelian stabilizer problem, [arXiv:quant-ph/9511026](https://arxiv.org/abs/quant-ph/9511026).
- [12] D. S. Abrams and S. Lloyd, Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors, *Phys. Rev. Lett.* **83**, 5162 (1999).
- [13] J.-H. Chen, Lower bounds for adiabatic quantum algorithms by quantum speed limits, *Phys. Rev. Res.* **5**, 033175 (2023).
- [14] S. Boixo and R. D. Somma, Necessary condition for the quantum adiabatic approximation, *Phys. Rev. A* **81**, 032308 (2010).
- [15] J. Roland and N. J. Cerf, Quantum search by local adiabatic evolution, *Phys. Rev. A* **65**, 042308 (2002).
- [16] A. T. Rezakhani, W.-J. Kuo, A. Hamma, D. A. Lidar, and P. Zanardi, Quantum adiabatic brachistochrone, *Phys. Rev. Lett.* **103**, 080502 (2009).
- [17] A. T. Rezakhani, D. F. Abasto, D. A. Lidar, and P. Zanardi, Intrinsic geometry of quantum adiabatic evolution and quantum phase transitions, *Phys. Rev. A* **82**, 012321 (2010).
- [18] S. Isermann, On the optimal schedule of adiabatic quantum computing, *Quantum Inf. Process.* **20**, 300 (2021).
- [19] S. Matsuura, S. Buck, V. Senicourt, and A. Zaribafiyani, Variationally scheduled quantum simulation, *Phys. Rev. A* **103**, 052435 (2021).
- [20] Y. Shingu and T. Hatomura, Geometrical scheduling of adiabatic control without information of energy spectra, *Phys. Rev. A* **112**, 022410 (2025).
- [21] M. Jarret, B. Lackey, A. Liu, and K. Wan, Quantum adiabatic optimization without heuristics, [arXiv:1810.04686](https://arxiv.org/abs/1810.04686).
- [22] A. Braidia, S. Chakraborty, A. Chaudhuri, J. Cunningham, R. Menavlikar, L. Novo, and J. Roland, Unstructured adiabatic quantum optimization: Optimality with limitations, *Quantum* **9**, 1790 (2025).
- [23] S. Boixo, E. Knill, and R. D. Somma, Eigenpath traversal by phase randomization, *Quantum Inf. Comput.* **9**, 833 (2009).
- [24] J.-F. Chen, Speeding up quantum adiabatic processes with a dynamical quantum geometric tensor, *Phys. Rev. Res.* **4**, 023252 (2022).
- [25] B. O'Neill, *Elementary Differential Geometry*, 2nd ed. (Academic Press, Burlington, MA, 2006).
- [26] P. M. Alsing and C. Cafaro, From the classical Frenet-Serret apparatus to the curvature and torsion of quantum-mechanical evolutions. Part II. Nonstationary Hamiltonians, *Int. J. Geom. Methods Mod. Phys.* **21**, 2450151 (2024).
- [27] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/ygs3-xgb1> for additional proofs and numerical data.
- [28] A. Elgart and G. A. Hagedorn, A note on the switching adiabatic theorem, *J. Math. Phys.* **53**, 102202 (2012).
- [29] A. D. Becke, Density-functional thermochemistry. III. The role of exact exchange, *J. Chem. Phys.* **98**, 5648 (1993).
- [30] C. Lee, W. Yang, and R. G. Parr, Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density, *Phys. Rev. B* **37**, 785 (1988).
- [31] L. K. Grover, Quantum mechanics helps in searching for a needle in a haystack, *Phys. Rev. Lett.* **79**, 325 (1997).
- [32] R. F. Stewart, Small Gaussian expansions of Slater-type orbitals, *J. Chem. Phys.* **52**, 431 (1970).
- [33] H. Beinert, R. H. Holm, and E. Münck, Iron-sulfur clusters: Nature's versatile electron-transfer agents, *Science* **277**, 653 (1997).
- [34] M. K. Johnson and A. D. Smith, Iron-Sulfur Proteins, in *Encyclopedia of Inorganic Chemistry*, edited by R. B. King, R. H. Crabtree, C. M. Lukehart, D. A. Atwood, and R. A. Scott (John Wiley & Sons, New Jersey, 2005).
- [35] M. Reiher, N. Wiebe, K. M. Svore, D. Wecker, and M. Troyer, Elucidating reaction mechanisms on quantum computers, *Proc. Natl. Acad. Sci. USA* **114**, 7555 (2017).
- [36] S. Sharma, K. Sivalingam, F. Neese, and G. K.-L. Chan, Low-energy spectrum of iron-sulfur clusters directly from many-particle quantum mechanics, *Nat. Chem.* **6**, 927 (2014).
- [37] S. Lee, J. Lee, H. Zhai, Y. Tong, A. M. Dalzell, A. Kumar, P. Helms, J. Gray, Z.-H. Cui, W. Liu, *et al.*, Evaluating the evidence for exponential quantum advantage in ground-state quantum chemistry, *Nat. Commun.* **14**, 1952 (2023).
- [38] https://github.com/mchan90/codes_for_published_paper/tree/main/CGS.
- [39] M. Han, H. Park, and S. Choi, Quantum Zeno Monte Carlo for computing observables, *npj Quantum Inf.* **11**, 46 (2025).
- [40] F. N. Fritsch and J. Butland, A method for constructing local monotone piecewise cubic interpolants, *SIAM J. Sci. Stat. Comput.* **5**, 300 (1984).
- [41] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, *et al.*, SciPy 1.0: Fundamental algorithms for scientific computing in Python, *Nat. Methods* **17**, 261 (2020).