

Counterdiabatic, Better, Faster, Stronger:
Overcoming Losses in Quantum Processes
PhD Thesis

Ieva Čepaitė

Quantum Optics and Quantum Many-Body Physics

Department of Physics

University of Strathclyde, Glasgow

June 2, 2023

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Abstract

Contents

Abstract	ii
Lay Summary	vi
Acknowledgements	vii
Acronyms and abbreviations	viii
List of Figures	viii
1 Introduction	2
1.1 Thesis overview	2
1.2 Publications and manuscripts	2
1.3 Talks and presentations	3
I Background	4
2 Quantum Adiabaticity	5
2.1 The quantum adiabatic theorem	6
2.1.1 Proof of the adiabatic theorem	7
2.1.2 The adiabatic condition: how slow is <i>slow</i> ?	11
2.2 The adiabatic gauge potential	13
2.2.1 The moving frame Hamiltonian	14
2.2.2 Matrix elements of the AGP	16
2.3 Counterdiabatic Driving	17

Contents

2.3.1	Brief interlude: the waiter and the glass of water	21
2.4	The approximate counterdiabatic drive	23
2.4.1	Local counterdiabatic driving	23
2.4.2	Nested commutator expansion	27
3	Quantum Optimal Control	29
3.1	The structure of optimal control problems	30
3.1.1	Mathematical structure	30
3.1.2	Analytic optimisation	34
3.1.3	Numerical optimisation	36
3.2	Quantum optimal control	45
3.3	Quantum optimal control methods	48
3.3.1	Chopped random-basis quantum optimization (CRAB)	48
3.3.2	Gradient Ascent Pulse Engineering (GRAPE)	51
II	Optimising approximate counterdiabatic driving	55
4	Counterdiabatic optimised local driving	56
4.1	Combining counterdiabatic driving and optimal control	57
4.2	The constraints	57
4.3	Optimal control toolbox	57
4.3.1	COLD-CRAB	57
4.3.2	COLD-GRAPE	57
5	Higher-order countediabatic driving as a cost function	59
5.1	The idea	59
III	Applications	60
6	Optimising for state fidelity	61
6.1	Two-spin annealing	61
6.2	Ising chain	62

Contents

– DRAFT – June 2, 2023 –

6.3	Transport in a synthetic lattice	62
6.4	Preparing GHZ states in a system of frustrated spins	63
7	Higher order AGP as a cost function	65
7.1	Return to two-spin annealing	65
7.2	Random Ising graphs	66
7.2.1	Maximal Independent Set	66
7.3	Spin-1/2 XXZ chains of ultracold atoms	66
IV	Conclusion	67
8	Conclusion	68
9	To boldly go	69
9.1	Exploring the information contained in the AGP	69
9.2	Applying COLD to useful problems	69
A	Rotating spin Hamiltonian	70
B	Derivation of the CD coefficients for an arbitrary Ising graph	71
C	More details and plots for the higher order AGP chapter	76
	Bibliography	76

Lay Summary

“With magic, you can turn a frog into a prince. With science, you can turn a frog into a Ph.D and you still have the frog you started with.”

Terry Pratchett

Here I'll basically just try to dump a short version of my blogpost [1].



Figure 1: a turtle with a jetpack strapped to its back, DALL·E

Acknowledgements

I would like to acknowledge that this was really really hard to put together and I need a break now.

Acronyms and abbreviations

AGP	Adiabatic Gauge Potential
ARP	Adiabatic Rapid Passage
BDA	Bare Dual Annealing
BPO	Bare Powell Optimisation
CD	Counterdiabatic driving
COLD	Counterdiabatic Optimised Local Driving
CRAB	Chopped Randomised Basis
GRAPE	Gradient Ascent Pulse Engineering
GSA	Generalized Simulated Annealing
LCD	Local Counterdiabatic Driving
PMP	Pontryagin Maximum Principle
QOCT	Quantum Optimal Control Theory
STA	Shortcuts to Adiabaticity

List of Figures

1	DALL-E turtle	vi
2.1	Rotating spin Bloch sphere illustrations	7
2.2	Rotating spin Bloch sphere illustration with counterdiabatic driving	26
3.1	Illustration of optimal control problem structure	32
3.2	Visualising the Nelder-Mead optimisation algorithm.	39
3.3	Visualising optimisers in action	42
3.4	Schematic diagram of open-loop quantum optimal control	46
6.1	COLD applied to two-spin annealing	62
7.1	Two-spin annealing fidelity contour plots	65
C.1	Placeholder: integral unscaled	77
C.2	Placeholder: max amplitude unscaled	77
C.3	Placeholder: max amplitude scaled by Hamiltonian norm	77

Chapter 1

Introduction

The fundamental goal of quantum technologies is the ability to manipulate and interact with quantum systems in a way that generates useful information, whether for the purpose of

1.1 Thesis overview

The structure of this thesis is as follows: in Chapter 2

Reminder: include a figure to illustrate the structure of the thesis, since this is clearly easy enough to do

Reminder: Here be the context for quantum adiabaticity, include stuff about STA etc.

1.2 Publications and manuscripts

The majority of this work is based on the following publications and manuscripts:

- (1) **Counterdiabatic Optimised Local Driving**, *Ieva Čepaité, Anatoli Polkovnikov, Andrew J. Daley, Callum W. Duncan. PRX Quantum 4, 010309, 2023. Eprint arxiv:2203.01948. [2]*
- (2) **Many-body spin rotation by adiabatic passage in spin-1/2 XXZ chains of ultracold atoms**, *Ivana Dimitrova, Stuart Flannigan, Yoo Kyung Lee, Hanzhen*

Lin, Jesse Amato-Grill, Niklas Jepsen, Ieva Čepaitė, Andrew J. Daley, Wolfgang Ketterle. Quantum Sci. Technol. 8 035018, 2023 Eprint arxiv:2301.00218. [3].

- (3) **Numerical approaches for non-adiabatic terms in quantum annealing,**
Ewen D. C. Lawrence, Sebastian Schmid, Ieva Čepaitė, Peter Kirton, Callum W. Duncan Eprint arxiv:0000.00000.

1.3 Talks and presentations

- “*Solving Partial Differential Equations (PDEs) with Quantum Computers*”, Atomic Weapons Establishment, (March 2020)
- “*A Continuous Variable Born Machine*”, Pittsburgh Quantum Institute Virtual Poster Session, Online (April 2020)
- “*A Continuous Variable Born Machine*”, Quantum Techniques in Machine Learning, Online (November 2020)
- “*Variational Counterdiabatic Driving*”, University of Strathclyde and University of Waterloo Joint Virtual Research Colloquium on Quantum Technologies, Online (November 2020)
- “*A Continuous Variable Born Machine*”, Bristol QIT Online Seminar Series, Online (March 2021)
- “*Optimised counterdiabatic driving with additional terms*”, APS March Meeting, Online (March 2021)
- “*Counterdiabatic Optimised Local Driving*”, DAMOP, Orlando (May 2022)
- “*Counterdiabatic Optimised Local Driving*”, QCS Hub Project Forum, Oxford (January 2023)
- “*Counterdiabatic Optimised Local Driving*”, APS March Meeting, Las Vegas (March 2023)
- “*Counterdiabatic Optimised Local Driving*”, INQA Seminar, Online (March 2023)

Part I

Background

Chapter 2

Quantum Adiabaticity

“I saw this movie about a bus that had to SPEED around a city, keeping its SPEED over fifty, and if its SPEED dropped, it would explode! I think it was called ‘The Bus That Couldn’t Slow Down’.”

Homer Simpson, *The Simpsons*
(S7E10)

The concept of quantum adiabaticity is the central starting point of the work presented in this thesis. While in classical thermodynamics, an adiabatic process is essentially one where no heat or mass is transferred between a system and its environment, the quantum adiabatic theorem concerns itself more with the speed at which changes in a time-dependent system Hamiltonian occur and how that speed affects the driven system. To illustrate, imagine a system that starts in some eigenstate of a Hamiltonian. If a parameter of the Hamiltonian is varied slowly enough, then the system is expected to stay in the corresponding eigenstate of the time-independent ‘snapshot’ Hamiltonian throughout the change and the process is ‘adiabatic’. In Sec. 2.1 I will derive the adiabatic condition and explore what happens when the rate of change in the Hamiltonian parameters is too fast for adiabaticity. As we will find, the non-adiabatic effects that

Chapter 2. Quantum Adiabaticity

result from fast driving are geometric in nature and related to an operator known as the adiabatic gauge potential [4, 5] or AGP, which I will describe in detail in Sec. 2.2 and proceed to use it in order to define the concept of counterdiabatic driving [6, 7] (CD) in Sec. 2.3. CD is a method under the more general umbrella of Shortcuts to Adiabaticity [8] (STA), which aim to suppress the non-adiabatic eigenstate deformations that occur when the Hamiltonian parameters are changed too fast in order to achieve pseudo-adiabatic processes at shorter timescales. In Sec. 2.4, I will demonstrate that exact suppression of non-adiabatic effects in the general case turns out to be impractical (if not impossible) and discuss how one can construct approximate CD protocols which are physically implementable and can mitigate some level of the losses brought about by fast driving.

2.1 The quantum adiabatic theorem

Imagine a quantum system that begins in the non-degenerate ground state of a time-dependent Hamiltonian. According to the the quantum adiabatic theorem, it will *remain* in the instantaneous ground state provided the Hamiltonian changes sufficiently slowly (where the meaning of ‘slow’ will become clearer as this section progresses). To take an intuitive example, we can consider a spin in a magnetic field that is rotated from the x direction to the z direction during some total time τ . The Hamiltonian might be written as:

$$H(t) = -\cos\left(\frac{\pi t}{2\tau}\right)\sigma^x - \sin\left(\frac{\pi t}{2\tau}\right)\sigma^z, \quad (2.1)$$

with the Pauli matrices defined as:

$$\sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (2.2)$$

If the spin starts in the ground state of $H(0)$ (pointing in the x direction, $|\psi(0)\rangle = |+\rangle$), then as the magnetic field is rotated, the spin starts precessing about the new direction of the field. This moves the spin toward the z axis but also produces a component out of the xz plane. As the total time for the rotation gets longer (i.e. the rotation

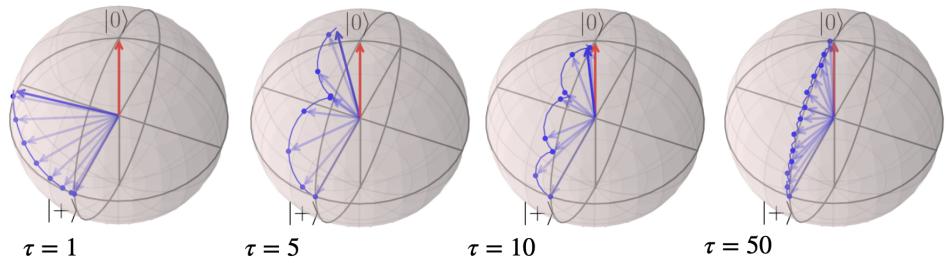


Figure 2.1: Bloch sphere illustration of the single-spin system driven by the Hamiltonian of Eq. (2.1) for different total driving times τ . The red arrow indicates the ground state of the Hamiltonian at $t = \tau$ while the blue path is that taken by the spin during the evolution.

gets slower compared to the precession), the state maintains a tighter and tighter orbit around the field direction. In the limit of $\tau \rightarrow \infty$, the state of the spin tracks the magnetic field perfectly, always in the ground state of $H(t)$ for all t . This is illustrated in Fig. 2.1, which shows the evolution of the system for increasing τ (and thus decreasing speed). At very fast times, e.g. when $\tau = 1$, the state of the spin veers away from the instantaneous ground state completely, while for $\tau = 50$, the evolution tracks the instantaneous ground state quite closely.

2.1.1 Proof of the adiabatic theorem

The above example gives some intuition for the behaviour of quantum systems as the time of evolution is slowed down, but it doesn't quite answer the question of what it means to be ‘slow enough’ in the general case, i.e. what one would refer to as the *adiabatic condition*. In order to characterise this regime, we first imagine a state $\psi(t)$ which evolves under some time-dependent Hamiltonian $H(t)$. For convenience, we redefine time through the parameter $\lambda = \frac{t}{\tau} \in [0, 1]$, such that $\psi(t), H(t) \rightarrow \psi(\lambda), H(\lambda)$ vary smoothly as a function of λ . This is often done to capture the fact that there may be a natural parameterisation of the changing Hamiltonian such as, for example, two different angles describing a varying magnetic field, which we may want to explore. The parameter space we build generally has some geometric properties that relate to non-adiabatic effects, so it becomes important to talk about these abstract parameters instead of time. But more on this later!

For each value of λ throughout the evolution, we have a time-independent ‘instantaneous’ Hamiltonian which can be diagonalised:

$$H(\lambda) |n(\lambda)\rangle = E_n(\lambda) |n(\lambda)\rangle, \quad (2.3)$$

where $E_n(\lambda)$ are the eigenenergies and $|n(\lambda)\rangle$ are the eigenvectors. The time-evolution of a system is given by the Schrödinger equation $i\hbar\partial_t |\psi(\lambda)\rangle = H(\lambda) |\psi(\lambda)\rangle$ and since the family of eigenvectors $|n(\lambda)\rangle$ constitute a basis at every value of λ , we can expand the system state as:

$$|\psi(\lambda)\rangle = \sum_n c_n(\lambda) e^{i\lambda^{-1}\theta_n(\lambda)} |n(\lambda)\rangle, \quad (2.4)$$

where $c(\lambda)$ are time-dependent coefficients through the parameter λ and

$$\theta_n(\lambda) = -\frac{1}{\hbar} \int_0^\lambda E_n(\lambda') d\lambda' \quad (2.5)$$

is commonly referred to as the *dynamic phase*.

Thus, the task is now to solve the time-dependent Schrödinger equation:

$$i\hbar\dot{\lambda} |\partial_\lambda\psi(\lambda)\rangle = H(\lambda) |\psi(\lambda)\rangle, \quad (2.6)$$

where ∂_λ is the partial derivative with respect to the parameter λ . We can then use the expansion Eq. (2.4), differentiate and take the inner product with some eigenstate $\langle m(\lambda)|$ to get:

$$\begin{aligned} i\hbar\dot{\lambda}\partial_\lambda \sum_n c_n e^{i\lambda^{-1}\theta_n} |n\rangle &= H \sum_n c_n e^{i\lambda^{-1}\theta_n} |n\rangle \\ \sum_n \left(\partial_\lambda c_n |n\rangle + c_n |\partial_\lambda n\rangle + i\dot{\lambda}^{-1} \partial_\lambda \theta_n c_n |n\rangle \right) e^{i\lambda^{-1}\theta_n} &= -\frac{i}{\hbar\dot{\lambda}} \sum_n E_n c_n e^{i\lambda^{-1}\theta_n} |n\rangle \\ \sum_n \left(\partial_\lambda c_n |n\rangle + c_n |\partial_\lambda n\rangle \right) e^{i\lambda^{-1}\theta_n} &= 0 \\ \partial_\lambda c_m &= - \sum_n c_n \langle m | \partial_\lambda n \rangle e^{i\dot{\lambda}^{-1}(\theta_m - \theta_n)} \end{aligned} \quad (2.7)$$

where the last two lines are a consequence of the fact that $i\dot{\lambda}^{-1} \partial_\lambda \theta_n(\lambda) = -\frac{i}{\hbar\dot{\lambda}} E_n(\lambda)$

and the the orthogonality of $|m\rangle$ and $|n\rangle$ when $m \neq n$. Note that I have removed the explicit dependence on λ for the sake of readability and to make writing this all out more bearable and I will continue with this convention for the rest of the chapter unless otherwise stated.

The above differential equation is exact and describes the evolution of the coefficients c_n , but it doesn't give much of a clue as to what 'slow' time evolution means with respect to the changes in the Hamiltonian. For that, we can express the term $\langle m|\partial_\lambda n\rangle$ in terms of the changing Hamiltonian. This is done by differentiating Eq. (2.3) with respect to time and then again taking the inner product with $\langle m|$ to get:

$$\begin{aligned} \dot{\lambda} \left(\partial_\lambda H |n\rangle + H |\partial_\lambda n\rangle \right) &= \dot{\lambda} \left(\partial_\lambda E_n |n\rangle + E_n |\partial_\lambda n\rangle \right) \\ \langle m|\partial_\lambda H|n\rangle + \langle m|H|\partial_\lambda n\rangle &= \partial_\lambda E_n \langle m|n\rangle + E_n \langle m|\partial_\lambda n\rangle \\ E_m \langle m|\partial_\lambda n\rangle - E_n \langle m|\partial_\lambda n\rangle &= -\langle m|\partial_\lambda H|n\rangle, \quad m \neq n \\ \langle m|\partial_\lambda n\rangle &= -\frac{\langle m|\partial_\lambda H|n\rangle}{E_m - E_n}, \quad m \neq n \end{aligned} \tag{2.8}$$

Plugging this into the final line of Eq. (2.7), we find that:

$$\partial_\lambda c_m + c_m \langle m|\partial_\lambda m\rangle = \sum_{n \neq m} c_n \frac{\langle m|\partial_\lambda H|n\rangle}{E_m - E_n} e^{i\dot{\lambda}^{-1}(\theta_m - \theta_n)}. \tag{2.9}$$

When the term on the RHS is small we can neglect it and the solution for the remaining differential equation of c_m is just:

$$c_m(\lambda) = c_m(0) e^{i\gamma_m(\lambda)}, \tag{2.10}$$

where:

$$\gamma_m(\lambda) = i \int_0^\lambda \langle m|\partial_{\lambda'} m\rangle d\lambda' \tag{2.11}$$

is the geometric (or Berry) phase [9–11]. It arises from the fact that if the Hamiltonian varies according to λ in a closed loop way, i.e. it returns to its starting point at the end of the evolution, the wavefunction might not (see Box 2.1.1 for more intuition).

Box 2.1.1| The Geometric Phase

Think Foucault’s pendulum, which changes its plane of swinging due to the Earth’s rotation around its own axis and does not necessarily return to its initial state after a full rotation. Both the appearance of the geometric phase in Eq. (2.10) and the changing plane of Foucault’s pendulum are consequences of the geometry or ‘curvature’ of the parameter space in which the dynamics occur and are related to concepts like parallel transport. To illustrate, we can absorb the geometric phase into the adiabatic eigenstates via the transformation

$$|m'\rangle = e^{i\gamma_m(\lambda)} |m\rangle = e^{-\int_0^\lambda \langle m|\partial_{\lambda'} m\rangle d\lambda'} |m\rangle, \quad (2.12)$$

and then take the derivative of the above expression with λ followed by taking the inner product with $\langle m'|$. This gives:

$$\langle m'|\partial_\lambda m'\rangle = 0, \quad (2.13)$$

which in other words just means that some change in the parameter λ produces an eigenvector that is orthogonal to the unchanged eigenstate. This turns out to be the condition which defines parallel transport along a curve in a curved space [12], as analogous to the classical example of Foucault’s pendulum. The choice of phases in Eq. (2.12) is generally referred to as the parallel transport gauge.

The constraint that the RHS of Eq. (2.9) be negligible is exactly the adiabatic condition, which can be seen by checking that $|c_m(\lambda)|^2 = |c_m(0)|^2$ in Eq. (2.10). What this means is that a state starting in a particular eigenstate $|m(\lambda)\rangle$ will remain in that state under these circumstances, e.g. for $c_m(0) = 1$ and $c_{m \neq n}(0) = 0$:

$$|\psi(\lambda)\rangle = e^{i\dot{\lambda}^{-1}\theta_m(\lambda)} e^{i\gamma_m(\lambda)} |m(\lambda)\rangle \quad (2.14)$$

the m^{th} eigenstate stays in the m^{th} eigenstate.

So to understand adiabaticity, we need to understand what conditions lead to the case where

$$\sum_{n \neq m} c_n \frac{\langle m|\partial_\lambda H|n\rangle}{E_m - E_n} e^{i\dot{\lambda}^{-1}(\theta_m - \theta_n)} \ll 1, \quad (2.15)$$

which is exactly what the next section sets out to do.

2.1.2 The adiabatic condition: how slow is *slow*?

The condition given by Eq. (2.15) contains terms relating both to the rate of change of the Hamiltonian with respect to λ (expressed in terms of matrix elements $\langle m|\partial_\lambda H|n\rangle$) and the energy gap between eigenstates $E_m - E_n$. It is not too hard to see that when the energy gaps are very large, these terms can be neglected. However, let us try to derive a more concrete and quantitative measure for ‘slowness’.

First, we can go back to the intermediate result from Eq. (2.9) and write it out as:

$$\partial_\lambda c_m = \sum_n c_n \langle m|\partial_\lambda n\rangle e^{i\dot{\lambda}^{-1}(\theta_m - \theta_n)}. \quad (2.16)$$

Since we want to focus on the RHS terms where $m \neq n$, we can remove the $m = n$ term by a change of variables:

$$d_m = c_m e^{\int_0^\lambda \langle m|\partial_\lambda m\rangle d\lambda} = c_m e^{-i\gamma_m} \quad (2.17)$$

and then, using Eq. (2.16), we find

$$\begin{aligned} \partial_\lambda d_m &= - \sum_n c_n \langle m|\partial_\lambda n\rangle e^{i\dot{\lambda}^{-1}(\theta_m - \theta_n)} e^{-i\gamma_m} + c_m \langle m|\partial_\lambda m\rangle e^{-\gamma_m} \\ &= - \sum_{n \neq m} d_n \langle m|\partial_\lambda n\rangle e^{-i(\gamma_m - \gamma_n)} e^{i\dot{\lambda}^{-1}(\theta_m - \theta_n)} \\ \Rightarrow e^{i\gamma_m} \partial_\lambda (c_m e^{-i\gamma_m}) &= - \sum_{n \neq m} c_n \langle m|\partial_\lambda n\rangle e^{i\gamma_n} e^{i\dot{\lambda}^{-1}(\theta_m - \theta_n)} \end{aligned} \quad (2.18)$$

Now all that is left is integration, which leads to:

$$c_m(1)e^{-i\gamma_m} = c_m(0) - \int_0^1 \sum_{n \neq m} c_n \langle m|\partial_\lambda n\rangle e^{i\dot{\lambda}^{-1}(\theta_m - \theta_n)} e^{i(\gamma_n - \gamma_m)} d\lambda. \quad (2.19)$$

In the above, we can see that when the integral on the RHS is 0, we recover the result in Eq. (2.10). The intuition is that when the integral is sufficiently small, the adiabatic condition is valid and the system will follow the instantaneous eigenstate. Since the integral is made up of a sum of terms of the same form, we can focus on

determining the bound on one of them. Representing the integral as:

$$I_{n \neq m}(1) = \int_0^1 c_n \frac{\langle m | \partial_\lambda H | n \rangle}{E_m - E_n} e^{i\dot{\lambda}^{-1}(\theta_m - \theta_n)} e^{i(\gamma_n - \gamma_m)} d\lambda, \quad (2.20)$$

where we used the result from Eq. (2.8). While the above integral doesn't inspire any confidence, it can be simplified a lot by using the fact that:

$$\begin{aligned} \partial_\lambda \left(c_n(\lambda) \frac{A_{m,n}(\lambda)}{\omega_{m,n}^2(\lambda)} e^{i\dot{\lambda}^{-1}(\theta_m - \theta_n)} \right) &= \partial_\lambda \left(c_n(\lambda) \frac{A_{m,n}(\lambda)}{\omega_{m,n}^2(\lambda)} \right) e^{i\dot{\lambda}^{-1}(\theta_m - \theta_n)} \\ &\quad - \frac{i}{\hbar \dot{\lambda}} c_n(\lambda) \frac{A_{m,n}(\lambda)}{\omega_{m,n}^2(\lambda)} e^{i\dot{\lambda}^{-1}(\theta_m - \theta_n)} \\ \Rightarrow c_n(\lambda) \frac{A_{m,n}(\lambda)}{\omega_{m,n}(\lambda)} e^{i\dot{\lambda}^{-1}(\theta_m - \theta_n)} &= -i\hbar \dot{\lambda} \left[\partial_\lambda \left(c_n(\lambda) \frac{A_{m,n}(\lambda)}{\omega_{m,n}^2(\lambda)} \right) e^{i\dot{\lambda}^{-1}(\theta_m - \theta_n)} \right. \\ &\quad \left. - \partial_\lambda \left(c_n(\lambda) \frac{A_{m,n}(\lambda)}{\omega_{m,n}^2(\lambda)} e^{i\dot{\lambda}^{-1}(\theta_m - \theta_n)} \right) \right], \end{aligned} \quad (2.21)$$

where, for the sake of sanity, readability and a decent reminder of the fact that the quantities above depend on λ , I have used $A_{m,n}(\lambda) = \langle m(\lambda) | \partial_\lambda H(\lambda) | n(\lambda) \rangle e^{-i(\gamma_m(\lambda) - \gamma_n(\lambda))}$ and $\omega_{m,n}(\lambda) = E_m(\lambda) - E_n(\lambda)$. This result can now be inserted into Eq. (2.20), leading to:

$$\begin{aligned} I_{n \neq m}(1) &= i\hbar \dot{\lambda} \left[c_n(\lambda) \frac{A_{m,n}(\lambda)}{\omega_{m,n}^2(\lambda)} e^{-\frac{i}{\hbar \dot{\lambda}} \int_0^\lambda \omega_{m,n}(\lambda') d\lambda'} \right]_0^1 \\ &\quad - i\hbar \dot{\lambda} \int_0^1 \frac{d}{d\lambda'} \left(c_n(\lambda) \frac{A_{m,n}(\lambda)}{\omega_{m,n}^2(\lambda)} \right) e^{-\frac{i}{\hbar \dot{\lambda}} \int_0^{\lambda'} \omega_{m,n}(\lambda'') d\lambda''} \\ &\approx -i\hbar \dot{\lambda} \left[c_n(1) \frac{A_{m,n}(1)}{\omega_{m,n}^2(1)} e^{-\frac{i}{\hbar \dot{\lambda}} \int_0^1 \omega_{m,n}(\lambda') d\lambda'} - c_n(0) \frac{A_{m,n}(0)}{\omega_{m,n}^2(0)} \right] \\ &= -i\hbar \dot{\lambda} c_n(1) \frac{A_{m,n}(1)}{\omega_{m,n}^2(1)} e^{-\frac{i}{\hbar \dot{\lambda}} \int_0^1 \omega_{m,n}(\lambda') d\lambda'}, \end{aligned} \quad (2.22)$$

where the last line is a consequence of the assumption that the system does not start in the eigenstate $|n \neq m\rangle$ and thus at $\lambda = 0$, the coefficient $c_{n \neq m} = 0$. As for the disappearing integral on the second line, this is due to the fact that $\dot{\lambda} = \frac{1}{\tau}$ and at long times $\tau \rightarrow \infty$, when the adiabatic condition is supposed to hold, the integrand will oscillate so fast that it will effectively vanish [13].

The term we're left with can effectively be bounded from above, since both exponential terms $e^{-\frac{i}{\hbar \dot{\lambda}} \int_0^1 \omega_{m,n}(\lambda') d\lambda'}$ and $e^{-i(\gamma_m(\lambda) - \gamma_n(\lambda))}$ (which has been absorbed into

Chapter 2. Quantum Adiabaticity

$A_{n,m}(\lambda)$) have a maximimal value of 1. The same goes for $c_n(1)$. This leaves us with a bound on the remaining quantities:

$$\max_{n,m} \left[\max_{\lambda} \left| \frac{\hbar \dot{\lambda} \langle m(\lambda) | \partial_{\lambda} H(\lambda) | n(\lambda) \rangle}{(E_m(\lambda) - E_n(\lambda))^2} \right| \right] \ll 1, \quad m \neq n, \quad (2.23)$$

which is exactly the adiabatic condition, as expected.

To illustrate the point more clearly, we can look back to the example Hamiltonian of Eq. (2.1), where the energy gap between the two eigenstates $|\psi_1(t)\rangle$ and $|\psi_2(t)\rangle$ is a constant: $E_{\psi_1} - E_{\psi_2} = 2$, and so are the matrix elements $\langle \psi_1 | \dot{H} | \psi_2 \rangle = \langle \psi_2 | \dot{H} | \psi_1 \rangle = \frac{\pi}{2\tau}$. The dependence on τ of the off-diagonal matrix elements of \dot{H} make the results of Fig. 2.1 immediately clearer: as τ increases (and hence the evolution is slower), the non-adiabatic component of Eq. (2.23) decreases proportionately to it. More details on the example and the derivation can be found in Appendix A.

In practice, it is not immediately obvious how the quantity stated in Eq. (2.23) relates to, say, the fidelity of the final state with respect to the desired state or how large τ , the evolution time, has to be in order to lead to a fidelity of some magnitude. While it is possible to find these bounds, the proof is quite lengthy and not necessary for the purposes of this thesis, so instead I will refer you to [14, 15] for more details.

2.2 The adiabatic gauge potential

The previous section introduced quantum adiabaticity and presented some intuition for non-adiabatic effects at fast driving times. In this section, I want to establish the deeply related concept of the adiabatic gauge potential (AGP) [4], a key player in the subject matter of this thesis and a fascinating mathematical object in its own right. While the AGP has primarily been studied in the context of suppressing non-adiabatic effects [16, 17], as will be its central role in this thesis, in recent years it has also been shown to be a potential probe for quantum chaos [18] and has been proposed for the study of thermalisation [19].

2.2.1 The moving frame Hamiltonian

In Section 2.1.1 we spent some time working in the instantaneous eigenbasis of the Hamiltonian where it is diagonalised, à la Eq. (2.3). For a general Hamiltonian, it is possible to go to this ‘moving frame’ picture by rotating the Hamiltonian via some unitary U so that it becomes diagonal at each point in time. If we start with some arbitrary Hamiltonian $H(\lambda)$ in some sort of ‘lab frame’ that depends on time through the parameter(s) $\lambda(t)$, it can be diagonalised through $\tilde{H} = U^\dagger(\lambda)H(\lambda)U(\lambda)$, where the \sim implies that we are now in the basis of the moving frame. In general, whenever the tilde symbol appears above an operator throughout this section, it means that we are working in this new basis: $\tilde{\sigma} = U^\dagger\sigma U$.

We can also view the quantum system evolving under the Hamiltonian in this moving frame picture: $|\tilde{\psi}\rangle = U^\dagger |\psi\rangle$, which is equivalent to expanding the wave function in the moving frame (or instantaneous) basis exactly as was done in Eq. (2.4). Given this new basisr, trying to solve the Schrödinger equation, this time reveals something interesting:

$$\begin{aligned} i\hbar \frac{d|\tilde{\psi}\rangle}{dt} &= i\hbar \left(\frac{dU^\dagger}{dt} |\psi\rangle + U^\dagger \frac{d|\psi\rangle}{dt} \right) \\ &= i\hbar \dot{\lambda} \frac{\partial U^\dagger}{\partial \lambda} |\psi\rangle + U^\dagger H |\psi\rangle \\ &= \dot{\lambda} \left(i\hbar \frac{\partial U^\dagger}{\partial \lambda} U \right) |\tilde{\psi}\rangle + U^\dagger H U |\tilde{\psi}\rangle \\ &= \left(\tilde{H} - \dot{\lambda} \tilde{\mathcal{A}}_\lambda \right) |\tilde{\psi}\rangle, \end{aligned} \tag{2.24}$$

where the operator $\tilde{\mathcal{A}}_\lambda$ is the *adiabatic gauge potential* with respect to the parameter λ in the moving frame of the Hamiltonian H and is expressed as:

$$\tilde{\mathcal{A}}_\lambda = i\hbar U^\dagger \partial_\lambda U, \tag{2.25}$$

The name ‘gauge potential’ generally refers to operators that are generators of continuous unitary translations in parameter space [4] of some unitary transformation U and generally takes the form of a derivative operator. For example, the gauge potential responsible for translations in space is just the momentum operator $p = i\hbar\partial_x$. This can be illustrated in the case of the simple 1D harmonic oscillator with a moving

potential centered on $x_0(t)$:

$$H(x_0) = \frac{p^2}{2m} + \frac{1}{2}m\omega^2(x - x_0)^2, \quad (2.26)$$

which can be diagonalised with the transformation $U(x_0) = e^{-ipx_0/\hbar}$. Then the gauge potential is simply:

$$\tilde{\mathcal{A}}_\lambda = i\hbar e^{ipx_0/\hbar} \partial_\lambda e^{-ipx_0/\hbar} = p. \quad (2.27)$$

While there are many such gauge potentials out there, we'll restrict ourselves to the specific case of adiabaticity where the transformation $U(\lambda)$ explicitly takes a wavefunction in an arbitrary basis to the adiabatic or instantaneous basis.

So what we find is that the wavefunction in the moving frame basis evolves under a combination of the diagonal component \tilde{H} and some additional term proportional both to the speed at which the parameter λ varies and the AGP. At this point I would like to simplify things by largely getting rid of the \sim , which we can do by applying the inverse unitary operation in order to work back in lab frame basis: $\tilde{H} - \dot{\lambda}\tilde{\mathcal{A}}_\lambda \xrightarrow{U\{\cdot\}U^\dagger} H - \dot{\lambda}\mathcal{A}_\lambda$. In fact, this transformation makes it easy to see how we can think of the AGP in the lab frame as nothing more than the derivative operator: $\mathcal{A}_\lambda = i\hbar\partial_\lambda$.

To see this, take any quantum state written in some basis, e.g. $|\psi\rangle = \sum_n \psi_n |n\rangle$. Then in the moving frame basis we have:

$$|\psi\rangle = \sum_n \psi_n U^\dagger(\lambda) |n\rangle = \sum_{\tilde{n}} \tilde{\psi}_n(\lambda) |\tilde{n}(\lambda)\rangle, \quad (2.28)$$

where $\tilde{\psi}_n(\lambda) = \sum_n U^\dagger(\lambda)\psi_n = \langle \tilde{n}(\lambda) | \psi \rangle$ and the dependence on λ enters into the basis vectors through the rotation $U(\lambda)$. These two bases can now allow us to look at the matrix elements of \mathcal{A}_λ in both:

$$\begin{aligned} \langle n | \tilde{\mathcal{A}}_\lambda | m \rangle &= \langle n | i\hbar U^\dagger \partial_\lambda U | m \rangle \\ &= i\hbar \langle \tilde{n}(\lambda) | \partial_\lambda | \tilde{m}(\lambda) \rangle \\ &= \langle \tilde{n}(\lambda) | \mathcal{A}_\lambda | \tilde{m}(\lambda) \rangle \end{aligned} \quad (2.29)$$

where the last line is simply the statement that in the lab frame $\mathcal{A}_\lambda = i\hbar\partial_\lambda$.

Chapter 2. Quantum Adiabaticity

2.2.2 Matrix elements of the AGP

What I've shown you so far is that when we try to solve the Schrödinger equation for a quantum system evolving under a time-dependent Hamiltonian in the basis of the moving frame, i.e. in the basis where the time-dependent Hamiltonian is diagonalised, we find that the evolution happens under a ‘decorated’ Hamiltonian composed of the diagonalised \tilde{H} and an additional operator generally known as the adiabatic gauge potential or AGP. We found that in the lab frame, it is the derivative operator with respect to the time-dependent parameters driving the Hamiltonian. What remains is to link this to our discussion of adiabaticity and the adiabatic condition of Section 2.1.

Let us return to the matrix elements of the lab frame AGP and see what they are in the adiabatic basis of Eq. (2.3), which is the eigenbasis of the instantaneous Hamiltonian. The first thing that jumps out is that the diagonal elements of the AGP are very familiar:

$$\langle n(\lambda) | \mathcal{A}_\lambda | n(\lambda) \rangle = i\hbar \langle n(\lambda) | \partial_\lambda | n(\lambda) \rangle. \quad (2.30)$$

The terms on the RHS are something known as the Berry connections and they look familiar because they are the integrands of the geometric phase (Eq. (2.11)) that we found when deriving the adiabatic condition. I mentioned that the geometric phase is related to the geometry or curvature of the parameter space of the adiabatic Hamiltonian and the AGP contains information about this geometry.

In order to understand what the off-diagonal elements of the AGP are, we can make use of the fact that in the instantaneous Hamiltonian basis $\langle m | H | n \rangle = 0$ for $m \neq n$. Differentiating with respect to the parameter λ gives:

$$\begin{aligned} \langle \partial_\lambda m | H | n \rangle + \langle n | \partial_\lambda H | m \rangle + \langle n | H | \partial_\lambda m \rangle &= 0 \\ E_n \langle \partial_\lambda m | n \rangle + E_m \langle m | \partial_\lambda n \rangle + \langle n | \partial_\lambda H | n \rangle &= 0 \\ (E_m - E_n) \langle m | \partial_\lambda n \rangle + \langle n | \partial_\lambda H | n \rangle &= 0 \\ \frac{-i}{\hbar} (E_m - E_n) \langle m | \mathcal{A}_\lambda | n \rangle + \langle n | \partial_\lambda H | n \rangle &= 0 \\ \langle m | \mathcal{A}_\lambda | n \rangle &= i\hbar \frac{\langle m | \partial_\lambda H | n \rangle}{(E_n - E_m)} \end{aligned} \quad (2.31)$$

Chapter 2. Quantum Adiabaticity

where, since we’re working in the adiabatic basis, all eigenstates, eigenenergies and the operators depend on λ . We can now see that \mathcal{A}_λ is Hermitian and the final line is familiar: the off-diagonal elements of the AGP are proportional to the non-adiabatic contribution we derived back in Eq. (2.22). The full operator in the adiabatic basis is:

$$\mathcal{A}_\lambda = i\hbar \left(\sum_n \langle n | \partial_\lambda n \rangle |n\rangle\langle n| + \sum_{m \neq n} |m\rangle \frac{\langle m | \partial_\lambda H | n \rangle}{(E_n - E_m)} \langle n | \right) \quad (2.32)$$

The outcome of this section then, is the revelation that this (initially mysterious) operator known as the AGP is deeply linked to the notion of adiabaticity in quantum systems: its diagonal terms are related to the geometry of the parameter space of adiabatic dynamics while its off-diagonals elements describe the non-adiabatic eigenstate deformations experienced by a state when it is driven by a time-dependent Hamiltonian. It is useful to note that in the final line of Eq. (2.24) we found that the Schrödinger equation corresponding to the evolution of the instantaneous eigenstates is:

$$i\hbar \frac{d|\tilde{\psi}\rangle}{dt} = (\tilde{H} - \dot{\lambda}\mathcal{A}_\lambda) |\tilde{\psi}\rangle, \quad (2.33)$$

where now it is not difficult to find how each of these operators contributes to the results of Section 2.1.1. The moving frame or instantaneous Hamiltonian generates the dynamical phase factor in Eq. (2.5), the diagonal elements of the AGP produce the geometric phase factor given by Eq. (2.11) and the off-diagonal elements of AGP are responsible for the non-adiabatic transitions out of the eigenstates which we upper bounded in Section 2.1.2. For a more detailed proof of how to derive the adiabatic theorem starting from Eq. (2.33) I recommend [20].

2.3 Counterdiabatic Driving

Having done all this work to characterise and understand the AGP and the adiabatic theorem, we now come to a very important question which defines just about the entirety of this thesis: *why do we care?* What is it about adiabatic dynamics and quantifying non-adiabatic transformations and understanding what generates them im-

Chapter 2. Quantum Adiabaticity

portant? The answer is simple: adiabatic processes are useful. The ability to drive a time-dependent Hamiltonian while remaining in a particular eigenstate can be used to prepare interesting quantum states [3], to solve combinatorics problems encoded in quantum systems [21, 22] or even to synthesise effective ramps and quantum logic gates [23] among many other applications.

The most natural way of exploiting adiabatic protocols is by adhering to the adiabatic condition. However, as is often the case when it come to the control of quantum systems, things aren't quite that simple. In practice, changing a Hamiltonian slowly enough to satisfy Eq. (2.23) leads to the system being overwhelmed by decoherence in many cases. Furthermore, as system sizes get larger, the energy gaps between the instantaneous eigenstates tend to get smaller requiring slower and slower driving, making adiabatic protocols unscalable. While the adiabatic condition is not impossible to adhere to, in order to have any hope of pushing quantum technologies beyond their current limits, it is necessary to move beyond the adiabatic limit. The result is that we need to find ways to achieve the same results as adiabatic processes but without requiring the prohibitively long driving times that are demanded by Eq. (2.23).

Our analysis of the adiabatic condition has given us a clue as to how we might be able to achieve fast driving without the eigenstate deformations that result from fast driving. Returning to Eq. (2.14), we may focus our attention on the fact that our goal is simply to have the system follow the eigenstates of the instantaneous Hamiltonian. The approach that aims to do exactly this was first developed independently by Demirplak and Rice [7] and Berry [6] and began as the observation that one can attempt to reverse-engineer a Hamiltonian that drives the instantaneous eigenstates exactly. Recall from Eq. (2.14) that in the case that we have adiabatic evolution, the instantaneous eigenstates evolve as $|\psi(\lambda)\rangle = e^{i\lambda^{-1}\theta_m(\lambda)}e^{i\gamma_m(\lambda)}|m(\lambda)\rangle$ with the dynamical phase θ_m and geometric phase γ_m defined in Eq. (2.5) and Eq. (2.11) respectively. If we want to find a Hamiltonian $H_{t\text{-less}}(\lambda)$ (transitionless) that drives these states exactly, we can

pick a unitary $R(\lambda)$ such that:

$$\begin{aligned} i\hbar\dot{\lambda}\partial_\lambda R(\lambda) &= H_{\text{t-less}}(\lambda)R(\lambda), \\ \Rightarrow H_{\text{t-less}}(\lambda) &= i\hbar\dot{\lambda}(\partial_\lambda R(\lambda))R^\dagger(\lambda). \end{aligned} \quad (2.34)$$

Turns out this unitary is just:

$$R(\lambda) = \sum_m e^{i\dot{\lambda}^{-1}\theta_m(\lambda)} e^{i\gamma_m(\lambda)} |m(\lambda)\rangle\langle m(0)|, \quad (2.35)$$

so the transitionless Hamiltonian can be expressed as:

$$\begin{aligned} H_{\text{t-less}}(\lambda) &= i\hbar\dot{\lambda} \sum_m \left[\left(-\frac{iE_m}{\dot{\lambda}\hbar} - \langle m(\lambda)|\partial_\lambda m(\lambda)\rangle \right) e^{i\dot{\lambda}^{-1}\theta_m(\lambda)} e^{i\gamma_m(\lambda)} |m(\lambda)\rangle\langle m(0)| \right. \\ &\quad \left. + e^{i\dot{\lambda}^{-1}\theta_m(\lambda)} e^{i\gamma_m(\lambda)} |\partial_\lambda m(\lambda)\rangle\langle m(0)| \right] e^{-i\dot{\lambda}^{-1}\theta_m(\lambda)} e^{-i\gamma_m(\lambda)} |m(0)\rangle\langle m(\lambda)| \\ &= \sum_m |m\rangle E_m \langle m| + i\hbar\dot{\lambda} \sum_m (|\partial_\lambda m\rangle\langle m| - \langle m|\partial_\lambda m\rangle |m\rangle\langle m|), \end{aligned} \quad (2.36)$$

where in the last line the dependence on λ has once again been removed from the eigenstates $|m\rangle$ and the eigenenergies E_m noting that all terms for $\lambda = 0$ have been cancelled out. In order to analyse the equation more easily, I'll rewrite it in terms of two separate components:

$$H_{\text{t-less}}(\lambda) = H_0(\lambda) + H_1(\lambda), \quad (2.37)$$

where

$$\begin{aligned} H_0 &= \sum_m E_m |m\rangle\langle m|, \\ H_1 &= i\hbar\dot{\lambda} \sum_m (|\partial_\lambda m\rangle\langle m| - \langle m|\partial_\lambda m\rangle |m\rangle\langle m|). \end{aligned} \quad (2.38)$$

What the above equation shows is that if we can engineer the Hamiltonian $H_{\text{t-less}}(\lambda)$, it is possible to drive the system arbitrarily fast, as it will always follow the instantaneous eigenstates. This might seem like a strange statement, but it becomes a lot simpler when we consider that the term H_1 looks quite familiar: it is nothing more

Chapter 2. Quantum Adiabaticity

than the negation of the AGP component in Eq. (2.33). The eagle-eyed among you might have already noticed that what Eq.(2.36) is actually doing in the moving frame is simply:

$$(\tilde{H} - \dot{\lambda}\tilde{\mathcal{A}}_\lambda) + \dot{\lambda}\tilde{\mathcal{A}}_\lambda = \tilde{H}, \quad (2.39)$$

which is reflected in the fact that H_1 is nothing more than the non-adiabatic contribution generated by the AGP. In the equation above, the effective Hamiltonian in the moving frame is simply the diagonalized version of the driving Hamiltonian $H(\lambda)$ in the lab frame, which does nothing more than drive the instantaneous eigenstates perfectly. This is the idea behind *counterdiabatic driving* or CD. The name, unsurprisingly, stems from the fact that the additional ‘counterdiabatic’ term $+\dot{\lambda}\mathcal{A}_\lambda$ is added in order to ‘counter’ the non-adiabatic or ‘diabatic’ effects that arise in the effective Hamiltonian throughout the system’s evolution.

Since all of the operators in the above equation are in the moving frame basis, we can rotate them back to the lab frame and explicitly define the counterdiabatic Hamiltonian:

$$H_{CD}(\lambda) = H(\lambda) + \dot{\lambda}\mathcal{A}_\lambda, \quad (2.40)$$

where we recall from Eq. (2.24) that $H(\lambda)$ becomes an effective $\tilde{H} - \dot{\lambda}\tilde{\mathcal{A}}_\lambda$ in the moving frame.

This seems simple enough: if H_{CD} is known and can be engineered, it is possible to drive a quantum system arbitrarily fast with no deformations associated with non-adiabatic effects. However, if this seems too good to be true, that’s because in general it is. The first clue is in the form of the AGP in Eq. (2.32), which implies that in order to know this CD Hamiltonian, we’d need to not only know the full eigenspectrum of the lab frame Hamiltonian for each value of λ throughout the protocol, but also to be able to engineer such terms to arbitrary precision in the lab. Furthermore, the off-diagonal elements of \mathcal{A}_λ , as alluded to earlier, are proportional to the inverse of the energy gaps in the system ($E_n - E_m$), which become exponentially small as system sizes increase and can become divergent or undefined in the thermodynamic limit [4, 5]. In chaotic systems, the AGP cannot be local because no local operator can distinguish many-

body states with arbitrary small energy difference [24]. What all of this really implies is that it is impossible - or at least impractical - to attempt to implement the exact counterdiabatic Hamiltonian given by Eq. (2.40) in the general case, barring some very simple and small systems.

2.3.1 Brief interlude: the waiter and the glass of water

It may seem like our inability to know or implement the exact CD Hamiltonian of Eq. (2.40) in the general case brings us back to square one in trying to speed up adiabatic protocols. I will show in the next section that it turns out this is not the case at all. However, before I dive back into the math, I wanted to take a moment to illustrate the concept of CD with a classical analogy which I first encountered in [16] and which I think not only elucidates what we've talked about so far, but also gives some intuition for how we might overcome the practical problems associated with the exact AGP, setting the stage nicely for the rest of the chapter.

The story goes something like this: imagine that you are a waiter tasked with carrying a glass of water on a tray from the bar to some table on the other side of a rather large restaurant. As you begin to walk while holding the tray perfectly level with the ground, your acceleration induces a force on the glass which causes it to wobble and the water to splash around. Ideally you would like to stop the water from spilling, so at this point you have two options: either to (a) walk slowly so as to minimize the force that is destabilizing the glass or else (b) to counteract it by tilting the tray (see Fig. [Reminder: add own figure, make it like a functor for each of the components!](#)).

You may already see where we are going with this. In the analogy, we can view the stable, upright state of the glass full of water as the ground state of some quantum system. The moving waiter then embodies the time-dependent Hamiltonian, where we can model their changing coordinates as they move through the bar via the abstract parameter(s) λ . Just as in the case of the adiabatic condition of Eq. (2.23), the chance of the glass tipping over depends on both the acceleration and direction of the waiter (the $\partial_\lambda H$ term) as well as how inherently stable the glass is due to e.g. a heavier bottom or more viscous liquid (the energy gap between the ground state and the nearest excited

state). In this picture, the two methods the waiter can use to stabilize the glass during transport are analogous to (a) following the adiabatic condition by minimizing the speed at which the Hamiltonian is deformed or (b) applying counterdiabatic driving i.e. counteracting the non-adiabatic force that appears as a consequence of their fast movement.

This example is not only useful for gaining intuition about adiabaticity and CD, but can also be used to bring attention to several interesting observations. The first is that by including a counterdiabatic component, the waiter introduces a new degree of freedom - a tilt - which would otherwise not show up anywhere in the process or the start/end points of the journey of the glass. Secondly, from the point of view of someone standing by the wayside (the lab frame), the glass is nowhere near standing upright throughout the counterdiabatic tilt, rather it is in some highly excited state, while from the perspective of the glass (the moving frame) it is quite stable and generally close to the instantaneous ground state, as can be garnered from looking at Equations (2.39) and (2.40) as representative of the two perspectives. The most important observation, however, which springboards us into the next section of this chapter, is precisely one which answers the question: *how stable is the glass throughout the waiter's counterdiabatic journey?* We cannot assume, in any realistic scenario, that the waiter has perfect knowledge of the movement of every molecule of water in the glass and can control their movements to such high precision that they instantly counteract even the smallest deviation from the perceived ground state. In fact, it is far more likely that the waiter has very limited ability to tilt the tray as well as only the roughest, low-resolution model of the ways in which the glass wobbles. The result is that far from implementing an ‘exact’ CD Hamiltonian as in Eq. (2.40), the waiter produces only some high-level approximation of the $\dot{\lambda}\mathcal{A}_\lambda$ term and yet, barring extreme circumstances, manages to quickly and safely transport the glass from bar to table.

2.4 The approximate counterdiabatic drive

Taking inspiration from the waiter story, we might imagine that a similar idea will hold true for CD protocols in the quantum setting. Why try to derive and implement the exact Hamiltonian of Eq. (2.40), when some rough version will cancel out most of the non-adiabatic effects? Even in our derivation of the adiabatic condition in Sec. 2.1.2 we upper-bounded the terms responsible for the unwanted transitions out of the eigenstate rather than trying to work with the full expression. It is this exact philosophy that is the backbone of the rest of this section, where I explore the different ways in which the AGP and thus the counterdiabatic drive can be approximated and what the advantages and drawbacks of each approach turn out to be.

2.4.1 Local counterdiabatic driving

The first method we will explore was developed by Sels and Polkovnikov in [16]: a variational minimization approach which I will refer to throughout this thesis as local counterdiabatic driving or LCD. Taking inspiration from the story of the waiter in the previous section, we can imagine constraining our counterdiabatic degrees of freedom in some way due to physical restrictions. In the case of the waiter, this might be related to the reaction time and physical capabilities of the human body, while in the case of quantum systems such degrees of freedom are generally best expressed as operators which may be implemented by some physical system. In the case of many-body systems, engineering arbitrary highly non-local many-body operators is hard and experimentally one tends to only have access to and control of only a limited set of physical operators. With this in mind, it makes sense to focus our approximation of the AGP to operators that are highly local or at least physically realisable, so that we could actually *implement* them when the time comes.

The task of not only finding a viable approximation of the CD, but also restricting it to a specific set of operators is not an easy one. Luckily, there is some structure in the AGP that we can exploit in order to write it in a slightly different form. We start

by differentiating the eigenenergies of the instantaneous basis Hamiltonian:

$$\begin{aligned}
 \frac{dE}{dt} &= \frac{d}{dt} \langle \tilde{\psi} | \tilde{H} | \tilde{\psi} \rangle \\
 &= \langle \partial_t \tilde{\psi} | \tilde{H} | \tilde{\psi} \rangle + \langle \tilde{\psi} | \partial_t \tilde{H} | \tilde{\psi} \rangle + \langle \tilde{\psi} | \tilde{H} | \partial_t \tilde{\psi} \rangle \\
 &= \frac{i}{\hbar} \langle \tilde{\psi} | (\tilde{H} - \dot{\lambda} \tilde{\mathcal{A}}_\lambda) \tilde{H} | \tilde{\psi} \rangle + \dot{\lambda} \langle \tilde{\psi} | \partial_\lambda \tilde{H} | \tilde{\psi} \rangle - \frac{i}{\hbar} \langle \tilde{\psi} | (\tilde{H} - \dot{\lambda} \tilde{\mathcal{A}}_\lambda) \tilde{H} | \tilde{\psi} \rangle \quad (2.41) \\
 &= \dot{\lambda} \langle \tilde{\psi} | \partial_\lambda \tilde{H} | \tilde{\psi} \rangle - \frac{i}{\hbar} \dot{\lambda} \langle \tilde{\psi} | [\tilde{\mathcal{A}}_\lambda, \tilde{H}] | \tilde{\psi} \rangle \\
 \Rightarrow i\hbar F_\lambda &= [\mathcal{A}_\lambda, H] - i\hbar \partial_\lambda H
 \end{aligned}$$

where we use the result from Eq. (2.24) that $i\hbar \partial_t |\tilde{\psi}\rangle = (\tilde{H} - \dot{\lambda} \tilde{\mathcal{A}}_\lambda) |\tilde{\psi}\rangle$ and

$$F_\lambda = - \sum_n \partial_\lambda E_n(\lambda) |n(\lambda)\rangle \langle n(\lambda)|, \quad (2.42)$$

is the generalised force operator [4, 16, 25].

The next step in the process is to define an operator G_λ :

$$G_\lambda(\bar{\mathcal{A}}_\lambda) = \partial_\lambda H + \frac{i}{\hbar} [\bar{\mathcal{A}}_\lambda, H], \quad (2.43)$$

where G_λ is parameterised by some Hermitian operator $\bar{\mathcal{A}}_\lambda$ which acts as an ansatz AGP. We can see that when $\bar{\mathcal{A}}_\lambda = \mathcal{A}_\lambda$, i.e. when our guess - or approximation - for the AGP is exactly correct, then $G_\lambda(\mathcal{A}_\lambda) = -F_\lambda$. This fact essentially allows us to reformulate the problem of trying to determine the AGP into one of minimization of distance between the operators $G_\lambda(\bar{\mathcal{A}}_\lambda)$ and $-F_\lambda$ with respect to the ansatz $\bar{\mathcal{A}}_\lambda$.

There are several options for a distance metric between two operators, but the task can be simplified simply by noticing that in the case where the ansatz is exact, G_λ has no off-diagonal elements, or $[H, G_\lambda(\mathcal{A}_\lambda)] = 0$. Thus a way to minimise the distance between $G_\lambda(\bar{\mathcal{A}}_\lambda)$ and $-F_\lambda$ is simply to minimise its Hilbert-Schmidt norm. Let us express this norm as an *action* [4] associated with the AGP:

$$\mathcal{S}(\bar{\mathcal{A}}_\lambda) = \text{Tr}[G_\lambda^2(\bar{\mathcal{A}}_\lambda)], \quad (2.44)$$

which is minimised whenever $\bar{\mathcal{A}}_\lambda$ satisfies:

$$\frac{\delta \mathcal{S}(\bar{\mathcal{A}}_\lambda)}{\delta \bar{\mathcal{A}}_\lambda} \Big|_{\bar{\mathcal{A}}_\lambda = \mathcal{A}_\lambda} = 0 \quad \Rightarrow \quad \left[H, \partial_\lambda H + \frac{i}{\hbar} [\mathcal{A}_\lambda, H] \right] = 0. \quad (2.45)$$

This all leads to a relatively simple recipe for finding a local, physically realisable counterdiabatic drive. To do this, we can choose a set of operators $\{\mathcal{O}_{LCD}\}$ which satisfy the constraints of our physical system. We can then define an approximate AGP in the basis of these operators as:

$$\bar{\mathcal{A}}_\lambda = \sum_j \alpha_j(\lambda) \mathcal{O}_{LCD}^{(j)}, \quad (2.46)$$

where the index j indicates the j^{th} operator in the basis and the coefficients $\alpha_j(\lambda)$ describe the continuous schedule of the counterdiabatic drive. Once we choose a set of operators $\{\mathcal{O}_{LCD}\}$, we can think of them as a fixed parameter, and the minimisation procedure consists of minimising the resulting action $\mathcal{S}(\bar{\mathcal{A}}_\lambda)$ with respect to the coefficients $\alpha_j(\lambda)$.

To make this clearer, let us return to the rotating spin Hamiltonian from Eq. (2.1). In order to simplify things, we can rewrite it with a change in parameters, taking $\lambda(t) = \frac{\pi t}{2\tau}$:

$$H(\lambda) = -\cos(\lambda)\sigma^x - \sin(\lambda)\sigma^z. \quad (2.47)$$

Since this is such a simple example, the only operators we could possibly include in the basis for our approximate AGP are single-spin operators. While any of the single-spin Pauli operators $\{\sigma^x, \sigma^y, \sigma^z\}$ are viable choices here, we note that it is not hard to see from $\tilde{\mathcal{A}}_\lambda = i\hbar U^\dagger \partial_\lambda U$ that if the Hamiltonian $H(\lambda)$ is real, the counterdiabatic term should be purely imaginary, as follows from the fact that a real Hamiltonian can always be diagonalised by a real orthogonal matrix U . This leaves us with a single degree of freedom that could act as the basis of $\bar{\mathcal{A}}_\lambda$, which is σ^y :

$$\bar{\mathcal{A}}_\lambda = \alpha(\lambda)\sigma^y. \quad (2.48)$$

In fact, as there are no other operators in this basis that could fit the description of being both a single-spin operator *and* imaginary, we expect that this ansatz should, for the correct α , be equal to the exact CD.

All that remains is to find $G_\lambda(\bar{\mathcal{A}}_\lambda)$ and to minimize the corresponding action $\mathcal{S}(\bar{\mathcal{A}}_\lambda)$ with respect to the driving coefficient α . Once this process is complete (see Appendix A for details), we find that

$$\alpha(\lambda) = -\frac{\sin^2(\lambda) + \cos^2(\lambda)}{2(\sin^2(\lambda) + \cos^2(\lambda))} = -\frac{1}{2}, \quad (2.49)$$

meaning that the counterdiabatic Hamiltonian can be written simply as:

$$\begin{aligned} H_{CD}(\lambda) &= H(\lambda) + \dot{\lambda}\alpha(\lambda)\sigma^y \\ &= -\cos(\lambda)\sigma^x - \sin(\lambda)\sigma^z - \frac{\pi}{4\tau}\sigma^y, \end{aligned} \quad (2.50)$$

where we have used the fact that $\dot{\lambda} = \pi/2\tau$. In Fig.2.2, we can see that even at very fast driving times, the rotating spin does not stray from the plane of rotation when the CD is applied. We can compare this to Fig. 2.1, where similar dynamics without the application of a CD drive were only achieved at around 500 times longer driving speeds.

In this case, it turns out that the counterdiabatic term is constant as a result of the choice of basis and $H(\lambda)$. In general, however, this is not the case and the CD term depends on λ through the coefficients of the lab frame Hamiltonian. Furthermore, while for this example only one operator was needed to describe the full CD, the number of such possible operators for a many-body system grows exponentially with system size, meaning that restricting to a highly-local and physically realisable basis makes for quite a sizeable reduction in the true number of possible operators in the full AGP.

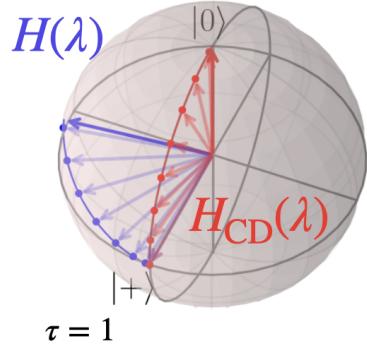


Figure 2.2: State of the rotating spin starting in state $|+\rangle$ driven without CD as in the Hamiltonian of Eq. (A.1) (blue) and with CD as given by Eq. (2.50) (red) for total driving time $\tau = 1$.

2.4.2 Nested commutator expansion

The LCD approach is fantastic in the case where one wants to implement a CD approximation constrained by some very limited, pre-determined set of operators, but it says absolutely nothing about what the operators should be when no constraints are imposed. The question I'd like to pose is whether or not there is any way to know what the operator basis of the approximate CD should be prior to performing the optimisation. This is useful not only in the case of determining the form of the CD in order to implement it, but also as a general tool in characterising non-adiabatic effects.

In this section I will focus on an approach developed in [17], where it was found that the AGP to some ℓ^{th} order can be extracted from a series of nested commutators:

$$\bar{\mathcal{A}}_{\lambda}^{(\ell)} = i\hbar \sum_{k=1}^{\ell} \alpha_k(\lambda) \underbrace{[H(\lambda), [H(\lambda), \dots [H(\lambda), \partial_{\lambda} H(\lambda)]]]}_{2k-1}, \quad (2.51)$$

where the coefficients $\alpha_k(\lambda)$ are used in a similar manner as LCD and can be found for each set of operators at order k using the minimisation procedure outlined in section 2.4.1. In the limit of $\ell \rightarrow \infty$, the expression above should represent the exact AGP, although there is no guarantee of convergence prior to this point as during each iteration of the commutations, the set of operators that are obtained need not be orthogonal to the previous set. This means that even when the set of operators contains all the required degrees of freedom to describe the exact AGP, the minimisation procedure is hampered due to the lack of linearity that is readily available in the LCD approach.

As noted in [17], there are several ways to motivate this form of the AGP, e.g. by noticing that such commutator terms appear in the Baker-Campbell-Hausdorff (BCH) expansion in the definition of a (properly regularized) [5] AGP for a fixed λ :

$$\mathcal{A}_{\lambda} = \lim_{\epsilon \rightarrow 0^+} \int_0^{\infty} dt e^{-\epsilon t} \left(e^{-iH(\lambda)t} \partial_{\lambda} H(\lambda) e^{iH(\lambda)t} + F_{\lambda} \right), \quad (2.52)$$

where F_{λ} is defined in Eq. (2.42). From the BCH expansion, we can find

$$e^{-iHt} \partial_{\lambda} H e^{iHt} = \sum_{k=0}^{\infty} \frac{(-it)^k}{k!} \underbrace{[H, [H, \dots [H, \partial_{\lambda} H]]]}_k, \quad (2.53)$$

where even-order commutators contribute to F_λ and odd-order commutators to \mathcal{A}_λ .

To gain more intuition for Eq. (2.51), one can try to evaluate it in the instantaneous eigenbasis of $H(\lambda)$:

$$\begin{aligned}\langle m | \bar{\mathcal{A}}_\lambda^{(\ell)} | n \rangle &= i\hbar \sum_{k=1}^{\ell} \alpha_k(\lambda) \langle m | \underbrace{[H(\lambda), [H(\lambda), \dots [H(\lambda), \partial_\lambda H(\lambda)]]]}_{2k-1} | n \rangle \\ &= i\hbar \left[\sum_{k=1}^{\ell} \alpha_k(\lambda) (E_m - E_n)^{2k-1} \right] \langle m | \partial_\lambda H | n \rangle,\end{aligned}\tag{2.54}$$

where we can see that the term we get out at the end looks very similar to the matrix elements we got in deriving the AGP in Eq. (2.31). In the case of the nested commutator expansion then, the use of the variational LCD approach in determining the coefficients α_k is equivalent to trying to approximate the factor $(E_m - E_n)^{-1}$ in the exact AGP via a power-series approximation:

$$\alpha_\lambda^{(\ell)}(\omega_{mn}) = \sum_{k=1}^{\ell} \alpha_k \omega_{mn}^{2k-1},\tag{2.55}$$

where $\omega_{mn} = (E_m - E_n)$. While this shows that the nested commutator approximation wouldn't work in regimes where the energy gap is exponentially small or exponentially big (i.e. where $\omega_{mn} \rightarrow 0$ or $\omega_{mn} \rightarrow \infty$), this turns out to not be an issue in practice. In the limit of very large energy gaps, the term $\langle m | \partial_\lambda H | n \rangle$ decays exponentially meaning that the contribution from these elements to the AGP is negligible anyway. As the energy gaps close, the AGP elements become undefined and generally in speeding up adiabatic processes, one only cares about suppressing transitions across some energy gap Δ . In that case, as long as $\omega_{mn} \geq \Delta$, the approximation does its job in the CD protocol.

In [17], it was shown that the approximate CD Hamiltonians constructed using $\mathcal{A}_\lambda^{(\ell)}$ can be implemented using Floquet Hamiltonians [26] by shaking the H and $\partial_\lambda H$ terms at frequencies determined by the coefficients α_k . However, for the purposes of this thesis, only the result presented in Eq. (2.51) matters, as far as it allows one to identify operators in the AGP basis without requiring to make an ansatz blindly.

Chapter 3

Quantum Optimal Control

“Neo, sooner or later you’re going to realize, just as I did, that there’s a difference between knowing the path and walking the path.”

Morpheus, *The Matrix* (1999)

Reminder: Please let me know where I should add citations!

The future of quantum technologies depends on our ability to control quantum systems with precision and accuracy. It is a key factor in the realisation of, for example, quantum computers [27], communication systems [28] and quantum sensors [29], as well as being necessary in the exploration and understanding of fundamental physics. The research field which concerns itself with such control problems is generally known as Quantum Optimal Control Theory (QOCT) [30,31] and its primary objective is the development of techniques which allow for the construction and analysis of strategies, primarily electromagnetic field shapes, that manipulate quantum dynamical processes in the most efficient and effective way possible in order to achieve certain objectives. Common control objectives in the quantum setting can range from state preparation [32] and quantum gate synthesis [23], to protection against decoherence [33] and entanglement generation [28].

While the field of quantum optimal control is vast and would take me an entire book

to summarize, this chapter aims to give a broad overview of the topic highlighting its structure, mechanisms, and practical applications, in particular with respect to the methods that are relevant to the rest of the work presented in this thesis. In Sec. 3.1, I will begin by exploring the general structure of optimal control problems and showing how an abstract goal can be transformed into a quantitative formula that guides us towards some desired outcome satisfying a given control objective. Sec. 3.2 will give an overview of how optimal control is adapted in the quantum setting and the main idea behind QOCT, while Sec. 3.3 will focus on specific methods used for constructing and optimising driving pulses with quantum systems in mind.

3.1 The structure of optimal control problems

The idea of an optimal control problem is simple: envision a target you want to achieve, cast it into some form of quantitative or abstract mathematical formula and then use said formula to derive the ‘best’ path to get to said objective. The aim of the first part of this section is thus to broadly cover the mathematical structure of optimal control problems and then, in the second, to delve more into practical questions of controllability and the process of optimisation.

3.1.1 Mathematical structure

In general, an optimal control problem is composed of a set of state functions $X : \mathbb{R} \rightarrow \mathbb{R}^n$, and a set of time-dependent control functions $U : \mathbb{R} \rightarrow \mathbb{R}^m$ and the optimal control problem consists of finding $x \in X$ and $u \in U$ that minimise some functional $C : X \times U \rightarrow \mathbb{R}$ such that the constraint:

$$\dot{x} = f(x, u), \quad (3.1)$$

is satisfied almost everywhere. This is a very abstract description and just about any control problem can be expressed as a special case of this formulation [34]. To gain more intuition, we can imagine a more concrete example where, e.g. U and X are sets of continuous functions on the interval $[0, \tau]$ satisfying $x(0) = x_0$. In this scenario,

τ could be a time interval during which we want to drive the system from an initial state x_0 to a final state x_f using the control function $u(t)$, $t \in [0, \tau]$. The choice of functional C would have to capture the desired outcome of the protocol: that the state of the system after the driving $x(\tau)$ be equal to the target x_f . This can be done by choosing a distance metric that depends on only on the drive u and is minimised when $x(\tau) = x_f$, e.g.

$$C(u) = \|x(\tau) - x_f\|^2. \quad (3.2)$$

The functional C is often referred to in literature as the *cost* or *loss* function [35] as it encodes the quality of the final protocol with respect to the desired outcome of the protocol. In that sense, we can imagine adding constraints to the problem that may increase the ‘cost’ of the protocol output if they are not satisfied to some degree. For example, Eq. (3.2) can be modified to include additional terms:

$$C(u) = \gamma \|x(\tau) - x_f\|^2 + \int_0^\tau \|u(t)\|^2 dt, \quad (3.3)$$

where γ is a penalty term on the final state that scales its importance relative to the additional second term, which is analogous to the cost in the energy required to achieve the final state. This updated cost function can be read as introducing a competition between the quality of the final state and the amount of energy expended to get it there, mediated by the value of γ .

There are several different types of problem structures in optimal control centering on different constraints and targets (see Box 3.1.1). In this thesis, we will primarily focus on what are often called *Mayer-type* problems [34], where the initial state is specified $x(0) = x_0$ and the cost function is of the form

$$C(u) = \phi(x(\tau), \tau), \quad (3.4)$$

where ϕ is a smooth function and τ the total time of the protocol. These two constraints and the requirement given by Eq. (3.1) define the state function x uniquely and the problem is then to determine a control function u on the appropriate set $[0, \tau]$ which

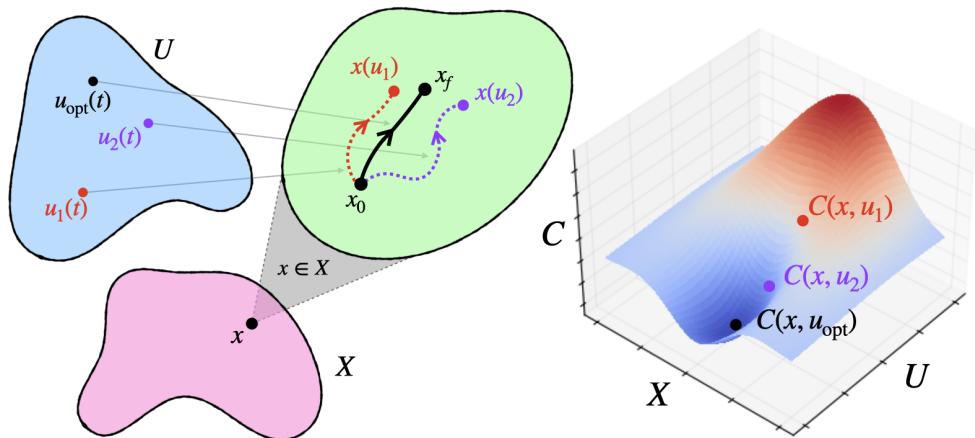


Figure 3.1: Illustration of the Mayer type optimal control problem: when an initial value of the system state x_0 is fixed, the choice of control function $u \in U$ and the requirement of satisfying Eq. (3.1) determine x uniquely. The task is then to find u_{opt} such that the functional $C(x, u_{\text{opt}})$ is minimised.

minimises Eq. (3.4). In Mayer-type problems, a specific target state can be defined in the cost function as a constraint, which is the case in Eq. (3.2) and this is illustrated in Fig. 3.1. However, this need not be the case as target states can be made implicit by having the cost function instead target some property of the state instead, like Euclidean distance from the initial state.

Apart from identifying the basic anatomy of control problems in terms of X , U and C , there is a myriad of additional information about their mathematical structure that can help to analyse and thus solve them. For example, it might be useful to identify if, for a particular optimal control problem, the system in question is *controllable* [36, 37] i.e. can any initial state be transformed into any desired target state. Equally, it might be useful to study what are called *reachable sets* [37, 38], which are sets containing all the states that an initial state can be driven to by the set of control functions U . In the case of Mayer-type problems, for example, it might be sensible to define a reachable set parameterised by the final evolution time τ such that it contains all possible states that can be obtained by the system during a driving time τ . Finally, I would be remiss not to mention the concept of *necessary conditions for optimality* [39], which focus on determining what formal conditions need to be satisfied for a specific control $u \in U$ to be optimal. Generally, this involves perturbing an assumed optimal control u by some

small parameter ϵ giving u^ϵ and then imposing the constraint that

$$C(u^\epsilon) - C(u) \geq 0, \quad (3.5)$$

which is then considered the necessary condition for optimality. The most basic of these optimality conditions is the Pontryagin maximum principle or PMP [40] (see Box 3.1.2), which states that for an optimal control problem, the optimal control and state trajectories should maximize a specific function which combines the system dynamics, the control inputs, and the Lagrange multipliers which encode the constraints of the control problem.

Box 3.1.1| Mayer, Lagrange and Bolza

There are primarily three types of optimal control problems: **Mayer**, **Lagrange** and **Bolza**. In the main text we cover the Mayer-type problem in detail, but it is useful to understand what differentiates it from the others.

- (1) Mayer-type problems are ones that are primarily concerned with the final state of the system and not its trajectory, with cost functions of the form given in Eq. (3.4).
- (2) Lagrange-type problems are ones where the focus is on the behaviour of the system throughout the trajectory and they encompass cost functions of the type

$$C(u) = \int_0^\tau L(x, u, t) dt \quad (3.6)$$

where L is a smooth function.

- (3) Bolza problems are a combination of the two, where both the system's behaviour during the trajectory and its final state matter:

$$C(u) = \phi(x(\tau), \tau) + \int_0^\tau L(x, u, t) dt, \quad (3.7)$$

where ϕ is a smooth function as given in Eq. (3.4). Bolza problems are thus more general and flexible, with the cost function in Eq. (3.3) serving as a great example.

3.1.2 Analytic optimisation

While the first part of optimal control is the construction of the problem, the second part is the search for a solution. The methods used to do this can generally be classified either as analytic or numerical approaches. While both are widely used in optimal control theory, this thesis will largely only focus on the latter, so I will be brief in introducing the former.

Analytic optimal control techniques are those that leverage mathematical rigor and formalism to derive solutions or insights, as opposed to relying primarily on numerical simulations, heuristics, or experimentation. They provide a theoretical foundation for understanding the properties and solutions of optimal control problems and are closely related to the discussion in Sec. 3.1.1. They can allow for a complete geometric understanding of the control problem leading to, for example, knowledge of the structure of a solution or even some proof about a global optimum. For a given set of constraints they might even be used to derive time limits of state transformations, i.e. the concept of reachability. An example of analytic methods is the aforementioned PMP, which provides information about the optimal solution by via a set of differential equations. A different analytic control theory tool, the Hamilton-Jacobi-Bellman equation [41], provides a way to find the optimal protocol via dynamic programming [42].

The trouble with analytic approaches, despite the commonplace rigorous guarantees of optimality and the scope of information they provide about the system, trajectory and structure of the control problems and their solutions, is that they are very difficult to scale up and quite inflexible to complex problem constraints. As such, analytic approaches are generally reserved for special cases, when problems have low dimensionality and simple structures where the cost function is generally linear in the arguments. Many real-world control systems require more complexity and flexibility than can be afforded by analytic methods.

Box 3.1.2| Pontryagin maximum principle**Theorem 3.1.1: PMP for Mayer problems**

For fixed final time τ and free final state assume u is the optimal control and x the corresponding trajectory solution of Eq. (3.1). Then, there exists a nonzero vector λ solution of the adjoint equations

$$\dot{\lambda}^T = -\lambda^T f(x(t), u(t)) \quad (3.8)$$

with terminal condition

$$\lambda^T(\tau) = -\phi(x(\tau)) \quad (3.9)$$

such that, for almost every $t \in (0, \tau]$, we have

$$\lambda^T(t)f(x(t), u(t)) \geq \lambda^T(t)f(x(t), v) \quad (3.10)$$

for every v in the set of the admissible values for the control U . Furthermore, for every $t \in [0, \tau]$

$$\lambda^T(t)f(x(t), u(t)) = c, \quad (3.11)$$

for a constant c

Using this, one can then define the *optimal control Hamiltonian*:

$$h(\lambda, x, u) := \lambda^T(t)f(x, u). \quad (3.12)$$

Now we can recast Eqs. (3.10) and (3.11):

$$\begin{aligned} h(\lambda(t), x(t), u(t)) &= c \\ h(\lambda, x, u) &\geq h(\lambda, x, v), \end{aligned} \quad (3.13)$$

The solution will be of the form $u := u(x, \lambda)$ and it can be solved with the system of equations

$$\begin{aligned} \dot{x} &= f(x, u(x, \lambda)), \\ \dot{\lambda}^T &= -\lambda^T f(x, u(x, \lambda)) \end{aligned} \quad (3.14)$$

with the boundary conditions $x(0) = x_0$ and $\lambda^T(\tau) = -\phi(x(\tau))$. Every control which is obtained with this procedure satisfies the necessary conditions of optimality and it is a candidate to be the optimal control.

3.1.3 Numerical optimisation

To overcome the drawbacks of analytic approaches, many optimal control problems are instead solved using numerical optimisation methods. These are generally algorithmic, iterative techniques which explore the cost function landscape step-by-step in order to converge to a minimum value. Numerical methods generally do not offer the same analysis or guarantees of optimality that analytic methods do. Their iterative nature may lead to a dependence of the outcome on the initial conditions of the algorithm, such as an initial guess for an optimal solution from which the iterations proceed or the bounds on the search space. Despite these drawbacks, however,

A general numerical optimisation technique consists of an initialisation step, a series of search steps and a termination step. These can be summarised as follows:

- (1) *Initialisation*: set up the necessary constraints of the optimal control problem, such as bounds on the solution space or an initial guess for the optimal solution.
- (2) *Search*: Perform some iterative search steps (deterministic or stochastic) with the goal of converging to the minimum of the cost function. What constitutes a single step varies massively between different techniques.
- (3) *Termination*: Return a solution after some condition is satisfied. This can be a convergence criterion based on the change in the cost function value between steps or a limit on the number of search steps that the algorithm is allowed to perform.

The simplicity of these three components leaves a lot of room for creativity and over the years many numerical optimisation algorithms and techniques have been developed to deal with different constraints and topologies of different cost function landscapes. It would take an entire book to cover the various categories and subcategories that exist within the field, so I will restrict myself to exploring a few key classifications of the structure of numerical optimisation methods.

One of the more broad ways to classify numerical optimisation methods is into the categories of *gradient-based* methods and *gradient-free* methods. Gradient-based

methods, as the name implies, make use of gradient information (the first derivative of the cost function) to guide the search for an optimal solution. These methods are often efficient and converge rapidly when the cost function is smooth and differentiable. A popular example of a gradient-based method is the gradient descent algorithm, which iteratively adjusts the solution in the direction opposite to the gradient, as this direction is likely the steepest decrease in the cost function value. A typical gradient descent protocol might look like:

$$\mathbf{u}_{t+1} = \mathbf{u}_t - \mu \nabla_{\mathbf{u}} C(\mathbf{u}), \quad (3.15)$$

where t denotes the current iteration of the algorithm, $\nabla_{\mathbf{u}} C(\mathbf{u})$ is the derivative of the cost function C with respect to the control parameters \mathbf{u} and μ is generally known as the ‘learning rate’ or ‘step size’ and its job is to control the resolution at which the algorithm traverses the cost landscape. Larger μ might lead to faster convergence but it might also mean overshooting the cost function minimum, so adjusting its value is often a heuristic that requires some experimentation. Other examples of gradient-based methods include Newton’s method and quasi-Newton methods [43], which employ information about the second derivative to guide the search and provide faster convergence as well as a myriad of other approaches including stochastic methods [44].

Gradient-free methods, on the other hand, do not require gradient information, making them suitable for optimization problems where the cost function is, e.g. discontinuous, non-differentiable, or its gradient is difficult or expensive to compute. Examples of gradient-free methods include particle swarm optimization [45], the Nelder-Mead method, which I will explore in more detail in Sec. 3.1.3.1 as well as the Powell method of Sec. 3.1.3.2. These methods often rely on trial and error, random sampling, or mimicking natural phenomena like evolutionary mechanisms [46] to explore the solution space. As in the case of gradient-based approaches, there is a veritable zoo of methods under this umbrella. As the rest of this thesis we will deal almost exclusively with gradient-free methods, I will provide examples of how these techniques look in the next couple of sections.

Box 3.1.3| Open-loop vs. closed-loop optimisation

In optimal control, there is a distinction between *open-loop* and *closed-loop* optimisation.

- *Open-loop* approaches calculate the control sequence ahead of time and apply it to the system irrespective of the system's actual behavior during the protocol.
- *Closed-loop* methods actively adjust the control strategy based on the current and past states of the system.

The closed loop approach is more resilient to uncertainties and disturbances but requires real-time computation or pre-computed feedback laws.

Apart from the gradient-information, another key way to classify optimisation algorithms is either as *local* or *global*. Local optimization methods are designed to find a local minimum, which is a solution that is better than all other feasible solutions in its vicinity in the landscape of the cost function. They are typically efficient at converging to the local minimum, but they provide no guarantee of finding the global minimum if the cost function is non-convex i.e. the local minimum is not automatically also the global minimum. Both Nelder-Mead and Powell are local methods.

Global optimization methods, on the other hand, aim to find a global optimum, which is the best solution among all feasible solutions, not just those in a local neighborhood. These methods typically employ a strategy to explore the entire solution space, either deterministically or stochastically, to avoid getting trapped in a local optimum. As a result of this larger scope, global optimization methods are generally more computationally intensive than local methods. An example of global optimisation that I will explore in more detail in Sec. 3.1.3.3 is Dual-Annealing, which combines generalized simulated annealing [47], a global search algorithm, with local optimisers in order to find an optimal solution. Global methods are often used when the optimization problem is complex, non-convex, or the global solution is significantly better than any local solution.

In the following sections I give examples of some common numerical optimisation methods that were used to obtain the results presented in this thesis.

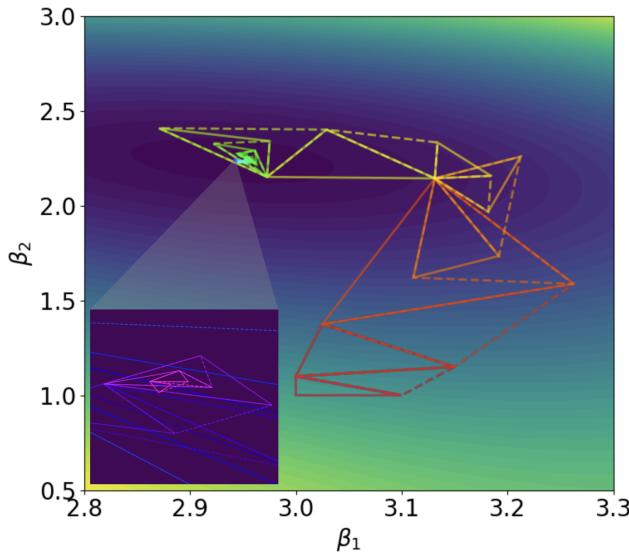


Figure 3.2: Illustration of the Nelder-Mead algorithm for a cost function parameterised by two parameters β_1 and β_2 . The simplex for a 2-dimensional landscape is a triangle, which then follows the rough algorithm described in the text. The iterations of the algorithm are indicated by constantly switching from solid to dashed edges of the simplex at each step as well as changing in colour. The inset shows a magnification of the final steps of the algorithm.

3.1.3.1 Nelder-Mead

A frequently used gradient-free optimisers is Nelder-Mead (or downhill-simplex) method [48] developed by J. Nelder and R. Mead in 1965. It is what's known as a *direct search* or *pattern search* approach and it is a gradient-free local method, making it generally quite efficient, but not guaranteed to converge to a global optimum of the cost function. Direct search methods work by varying each optimisable parameter by some small stepsize from the current minimum in each direction and computing the cost function at the updated value. The change that leads to the largest decrease in the cost function value is taken as the new minimum. Once no such variation leads to an improvement, the stepsize is halved and the process is repeated until some convergence criterion is satisfied.

The way this direct search approach is adapted in Nelder-Mead is by constructing simplices, which are geometric objects that generalise triangles in lower and higher dimensions. For a cost function dependent on n parameters, Nelder-Mead constructs an

Chapter 3. Quantum Optimal Control

n -dimensional simplex. For $n = 0$ this is a point, for $n = 1, 2, 3$ a line segment, triangle and tetrahedron respectively and then higher-dimensional versions as n increases. Thus a simplex has $n + 1$ vertices for n parameters.

The vertices of this simplex then traverse the cost function landscape according to the Nelder-Mead algorithm in order to converge to some minimum value. In most of the search steps, the primary change is to shift the highest point of the simplex (i.e., where the cost function value is largest) through the opposite face of the simplex, moving to a point with a lower cost function value. These steps are known as *reflections* and they are designed to preserve the volume of the simplex, ensuring it remains non-degenerate. Whenever possible, the method will expand the simplex along a particular direction, which allows it to take bigger steps in search of a minimum. When the simplex encounters a region that can be thought of as a ‘valley floor’ in the cost function landscape, it reduces its dimensions orthogonal to the valley, so that it can slide down. In situations where the simplex has to navigate through a narrow passage, it shrinks itself in all directions, wrapping itself around its best (lowest) point, enabling it to continue its search for the minimum. This process is illustrated for a simple example in Fig. 3.2.

This description of the Nelder-Mead method is quite rough and it only covers the basic idea that was first developed in the original 1965 paper. Many variations and improvements have been developed in the years since and the actual implementations. In general, the Nelder-Mead approach is simple to understand and implement, as well as being quite efficient and flexible. However, it often suffers from convergence issues, being both likely to return a sub-optimal local minimum and to get stuck without converging far longer than necessary, undoing any efficiency it otherwise promised. Furthermore, the simplex method doesn’t scale well in higher dimensions, making it less effective when the number of parameters is large.

3.1.3.2 Powell’s method

Another approach from the gradient-free, local optimiser crowd is Powell’s method, first developed by Michael J. D. Powell in 1964 [49]. The algorithm is known as a *conjugate-*

Chapter 3. Quantum Optimal Control

direction approach, not to be confused with the more common conjugate-gradient approach, although the two are related as the latter can be viewed as a specialisation of the former.

The basis of Powell’s method relies on the idea of conjugate vectors or conjugate directions. Two vectors \mathbf{u} and \mathbf{v} are said to be conjugate with respect to some positive semi-definite matrix A if $\mathbf{u}^T A \mathbf{v} = 0$. A set of conjugate directions, thus, is a set of vectors that are pairwise conjugate. Furthermore, one can make the observation [50] that the function

$$f(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} - 2\mathbf{b}^T \mathbf{x} + c \quad (3.16)$$

for some positive semidefinite matrix A , $\mathbf{b} \in \mathbb{R}^n$ and $c \in \mathbb{R}$ has a minimum at the point $\sum_{i=1}^n \beta_i \mathbf{u}_i$ in the space spanned by the set of conjugate vectors $\{\mathbf{u}_j\}_{j=1,\dots,n}$ with

$$\beta_i = \frac{\mathbf{u}_i^T \mathbf{b}}{\mathbf{u}_i^T A \mathbf{u}_i}. \quad (3.17)$$

This minimum can be calculated efficiently just through evaluating the function, without needing explicit access to A , \mathbf{b} or c . This property allowed Powell to develop a simple but powerful gradient-free approach, which can be summarised in the following bit of pseudocode.

Algorithm 1 Powell’s Method

```

1: procedure POWELL
2:   Initialise the method with ansatz solution  $\mathbf{u}_0 \in \mathbb{R}^m$  and  $n \leq m$  conjugate
   search vectors  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ . If none are provided, use columns of the  $m$ -dimensional
   identity matrix.
3:   for  $i = 1, \dots, n$  do
4:     Compute  $\beta_i$  to minimise  $f(\mathbf{u}_{i-1} + \beta_i \mathbf{x}_i)$ 
5:     Define  $\mathbf{u}_i \leftarrow \mathbf{u}_{i-1} + \beta_i \mathbf{x}_i$ 
6:   end for
7:   for  $i = 1, \dots, n - 1$  do
8:      $\mathbf{x}_i \leftarrow \mathbf{x}_{i+1}$ 
9:   end for
10:   $\mathbf{x}_n \leftarrow (\mathbf{u}_n - \mathbf{u}_0)$ 
11:  Compute  $\beta$  to minimize  $(f(\mathbf{u}_0 + \beta \mathbf{x}_n))$ 
12:   $\mathbf{u}_0 \leftarrow \mathbf{u}_0 + \beta \mathbf{x}_n$ 
13: end procedure

```

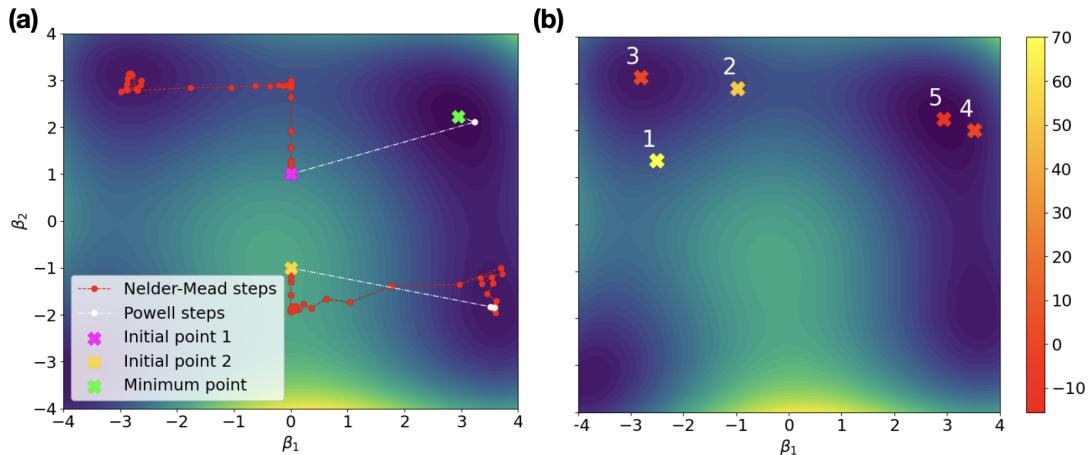


Figure 3.3: An illustration of the optimisation strategy of several numerical optimisation methods. (a) Local optimisers: steps of the Nelder-Mead method and Powell's method when instantiated in two different locations of the loss function landscape. The global minimum is illustrated by a green cross. (b) The local minima (crosses) visited by the Dual-Annealing in the order indicated by the numbered labels. The color of the crosses reflects the value of the cost function evaluated at that point as indicated by the colorbar.

In other words, the algorithm is initialised with a guess for a solution and a set of conjugate directions. It then proceeds to find a minimum along each direction and shifts to a new point along a superposition of their minima, adding the vector along in which it shifted to the list of conjugate direction vectors and removing the first vector in the list before starting the next step.

The Powell method is far more complex than is presented here, in particular due to the fact that several extra steps are usually added in order to guarantee convergence and additional features to help optimise it. Additionally, the minimisation procedure of steps 4 and 11 in the pseudocode is highly non-trivial and can be achieved via several different algorithms like Brent's method [50]. It has guarantees of being very efficient in convex optimisation problems and excels in high-dimensional spaces, unlike Nelder-Mead. A plot of the search steps of the two methods in Fig. 3.3(a) shows how they compare in terms of number of steps taken and accuracy in finding the optimum of some non-convex loss function. Importantly, given the more complicated nature of the steps in Powell's method, the fact that it requires less steps to converge to a solution

does not necessarily make it more efficient.

3.1.3.3 Dual-annealing

Unlike both Nelder-Mead and Powell's method, dual-annealing is a *global* optimization algorithm, meaning that its primary goal is to find a global minimum of the function. It is also a stochastic method, since rather than follow a pre-defined set of rules or procedures, it employs probabilistic transitions or decisions during the search. This added randomness can help the algorithm escape local optima and explore the solution space more broadly, however it also adds to the computational complexity of such approaches. As mentioned earlier, global optimisation algorithms tend to be far less efficient than local ones, but this is the price that needs to be paid when solutions obtained in local minima are simply not enough and the cost function landscape is highly non-convex.

What is particularly interesting about dual-annealing is that it combines Generalized Simulated Annealing (GSA) [47], a global search algorithm, with a choice of *local* optimiser that refines the solution once the global search is done. This is important because global algorithms, including GSA, are often good at locating the vicinity of the global minimum (the basin) but not necessarily the minimum itself.

The GSA part of dual-annealing function is, unsurprisingly, a generalisation of the simulated annealing algorithm [51] inspired by the annealing process of metallurgy which causes a molten metal to reach its crystalline state which is the global minimum in terms of thermodynamic energy. In simulated annealing, the cost function is treated as the energy function of a molten metal and one or more artificial temperatures are introduced and gradually cooled. In GSA, this presents itself as a series of probabilistic jumps across the cost function landscape that depend on an artificial temperature parameter which decreases as the search progresses.

More concretely, at each step of the search, the algorithm generates a trial jump in the cost function space from the current temporary solution to a new point by sampling it from a modified Cauchy-Lorentz distribution. The distribution peaks around the current temporary solution and its scale parameter (a variable that controls its spread)

is a function of the artificial temperature T_{qv} . Thus, the higher the temperature, the more likely it is that the trial jump will be larger. The q_v parameter can be set to different values in order to speed up or slow down the cooling process.

Once the trial jump has been generated, it is either accepted or rejected based on the cost function value at the new point as compared to the current point. If the new point is better (i.e. the jump is ‘downhill’, towards a lower energy), then the jump is accepted. If, on the other hand, the jump is worse or ‘uphill’, it might still be accepted with some probability based on a parameterised Metropolis algorithm [52]. This allows for the algorithm to potentially escape local minima. If a jump is accepted, the search then continues in a similar manner from the new point and the temperature parameter is decreased, reducing the probability of the next generated jump being far away from the current point.

The dual-annealing algorithm proceeds by first using GSA to identify a ‘basin’ in the cost function landscape and then using the best solution so far as an initial guess for a local optimisation algorithm like Nelder-Mead or Powell’s method to refine the solution. The local search is generally called when the artificial temperature decreases below some pre-defined value and once the local search is done, the whole process restarts again while keeping track of the current best solution. The entire algorithm terminates when some convergence criterion is satisfied. Usually this is when some number of search iterations or cost function evaluations is reached, or there is no more improvement to the solution below some tolerance. This process is illustrated in Fig. 3.3(b), where the dual-annealing algorithm returns points 1, 2 and 4 as minima detected during the annealing stages with 3 and 5 corresponding to minima detected during the local searches.

The verdict regarding dual-annealing, with respect to the local optimisers that we addressed previously, is that it is far more powerful and can lead to far better solutions, given that it has a far better ability to explore the cost function landscape. However, it is also more computationally expensive, as can be made obvious by the fact that local search is merely a subroutine of the algorithm. Ultimately, the choice of which approach comes down to having knowledge about the cost function landscape as well

Chapter 3. Quantum Optimal Control

as trial-and-error. The use of a global optimiser may be overkill when the cost function landscape lends itself well to local methods and each evaluation of the cost function is expensive. If, however, local optimal solutions are not enough, then global methods are by far the best option.

3.2 Quantum optimal control

We've now established that the broad goal of optimal control theory is the design of protocols and strategies which optimise the behaviour of some abstract control system with respect to some abstract target. Quantum optimal control theory (QOCT), rather predictably, does this in the setting where the abstract system is quantum. Very broadly then, QOCT concerns itself with the design and analysis of electromagnetic fields that manipulate quantum dynamical processes at the atomic or molecular scale in the best way possible, as illustrated in Fig. 3.4. In this thesis we will primarily focus on closed systems, where the generator of transformations is primarily modelled as the Hamiltonian, but a similar analysis holds in the case of open systems which are just as interesting, if not quite as relevant to my work.

Returning to the material covered in Sec. 3.1.1, we can now add more structure to the abstract notions of system, control function and cost function. In the quantum setting, the set of state functions X often takes the form of a set of quantum states, be they complex vectors, density matrices or operators. The set of control functions U is usually represented by a set of functions of parameterised Hamiltonians. It is common to decompose a control Hamiltonian into two components: the time-dependent ‘drive’ part and the time-independent ‘drift’ part. The time-dependent part can then be further decomposed into a set of N_k operators $\{\mathcal{O}_{\text{opt}}^{(k)}\}_{k=1,\dots,N_k}$, such that the full control Hamiltonian reads:

$$H(\mathbf{u}(t)) = H_0 + \sum_{k=1}^{N_k} u_k(t) \mathcal{O}_{\text{opt}}^{(k)}, \quad (3.18)$$

where H_0 is the drift Hamiltonian with no external controls and the control functions $u_k(t) \in \mathbf{u}(t)$ drive the corresponding operators $\mathcal{O}_{\text{opt}}^{(k)}$.

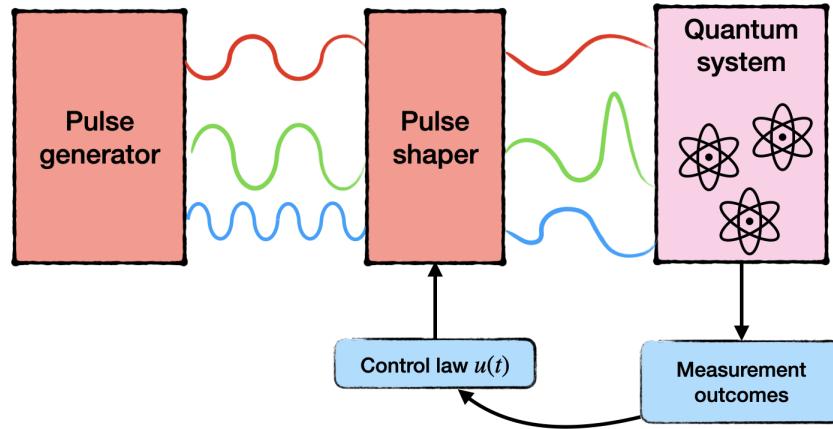


Figure 3.4: A sketch of a quantum optimal control open-loop set-up. A quantum system is directly controlled by a set of electromagnetic pulses which are shaped according to a set of control functions $\mathbf{u}(t)$ that are optimised based on feedback from the information obtained through measurements of the system.

Given this, we can describe a general quantum optimal control problem in analogy to Eq. (3.1) as one where the aim is to solve the Schrödinger equation:

$$i\hbar\partial_t |\psi(t)\rangle = H(\mathbf{u}(t)) |\psi(t)\rangle \quad (3.19)$$

with the constraint of starting in a state from a set of initial states $|\psi_0\rangle \in \Psi_0$ and while minimising some cost function that targets a set of final states $|\psi_T\rangle \in \Psi_T$. I should note that while I am using wavefunctions as representations for the states of the control system, it is actually quite common in QOCT to instead work in the operator picture, where the Schrödinger equation is

$$\partial_t \mathcal{O}(t) = -iH(\mathbf{u}(t)) \mathcal{O}(t), \quad (3.20)$$

where \mathcal{O} is an operator on some pre-defined Hilbert space. A useful constraint in this setting is to take the initial state of the operator to be the identity $\mathcal{O}(0) = \mathbb{1}$. The choice of wave mechanics or matrix mechanics depends on the specific QOCT problem at hand, although questions of e.g. controllability are usually best-solved with operators rather than state vectors. For example, if we can show that the set of possible matrices that

can be obtained for system (3.20) is the set of all the unitary matrices (with the rank of the system Hilbert space), then the system can theoretically be steered to any arbitrary state and thus it is controllable.

The choice of cost function in the quantum setting is generally informed by the desired properties of the target state(s) combined with considerations for what information can be extracted from the system and other constraints. For example, when the aim of the optimisation is to prepare a single, well-defined quantum state $|\psi_T\rangle$ with high accuracy, then the most informative cost function is:

$$C_F(\tau, \mathbf{u}) = 1 - F(\tau, \mathbf{u}) = 1 - |\langle\psi(\tau, \mathbf{u})|\psi_T\rangle|^2, \quad (3.21)$$

where $F(\tau, \mathbf{u})$ is the fidelity of the final state $|\psi(\tau, \mathbf{u})\rangle$ with respect to the desired target $|\psi_T\rangle$. Here $|\psi(\tau, \mathbf{u})\rangle$ is generated by driving an initial state $|\psi_0\rangle \in \Psi_0$ for a time τ via the time-dependent Hamiltonian $H(\mathbf{u}(t))$. If, on the other hand, the target state need only be a ground state of some Hamiltonian H_T , then it might be far more convenient to use the final system energy:

$$C_E(\tau, \mathbf{u}) = \langle\psi(\tau, \mathbf{u})|H_T|\psi(\tau, \mathbf{u})\rangle. \quad (3.22)$$

Finally, should one be interested only in a specific property of the final state, like its entanglement, then the cost function might look something like:

$$C_S(\tau, \mathbf{u}) = -S[|\psi(\tau, \mathbf{u})\rangle], \quad (3.23)$$

where $S[\cdot]$ is some appropriate measure of entanglement. As well as being informed by the set of target states, the cost function may include further constraints, like the total power of the driving fields in the Hamiltonian. This is analogous to the cost function in Eq. (3.3), which in the quantum case might look something like:

$$C(\tau, \mathbf{u}) = C_F(\tau, \mathbf{u}) + \sum_{k=1}^{N_k} \int_0^\tau |u_k(t)|^2 dt, \quad (3.24)$$

Chapter 3. Quantum Optimal Control

which can be read as an optimisation for final state fidelity with the added constraint that the time-integrated flux of the driving fields is minimised.

3.3 Quantum optimal control methods

Both analytic (Sec. 3.1.2) and numerical (Sec. 3.1.3) methods have been developed for the optimal control of quantum systems in recent decades. Analytic methods in QOCT generally deal with questions of necessary conditions for controllability [53] or reachability of states, e.g. exploring quantum speed limits [54]. Analytic methods *can* be used to find solutions to quantum optimal control problems rather than just classify their structure, but the Achilles’ heel of analytic approaches remains a general inability to deal with complex systems. The volatile and often exponentially complex nature of quantum systems means that numerical approaches tend to be the preferred method for actually determining solutions to QOCT problems.

Numerical methods in QOCT tend to consist of the development and analysis of iterative algorithms focused on optimising pulses for quantum systems. This can be done by constructing a mathematical description of the pulse, including parameters that control its shape and which can then be numerically optimised. Most numerical methods under the umbrella of QOCT involve a classical optimiser, like those discussed in Sec. 3.1.3, as a subroutine in the approach which finds the optimal values for the pulse parameters. In this section I will explore two of the more broadly used numerical approaches in quantum optimal control, CRAB and GRAPE.

3.3.1 Chopped random-basis quantum optimization (CRAB)

The “Chopped random-basis quantum optimization” or CRAB method is a quantum optimal control method first introduced in [55, 56] which revolves around the construction of a truncated randomized basis of functions for the control fields of a quantum system. It was originally developed for quantum many-body systems whose time evolution can be efficiently simulated by time-dependent density matrix renormalization group (tDMRG) [57–59]. It was believed that such systems were mostly intractable for

control optimization using gradient-based algorithms [60], although such potential limitations have been overcome in more recent work [61]. CRAB provides a way to reduce the space of search parameters, making the optimisation process more efficient, while retaining access to a large solution space through the added randomisation component.

The key idea is to expand the control pulse $\mathbf{u}(t)$ in some truncated basis of dimension N_k :

$$\mathbf{u}(t) = \sum_{i=1}^{N_k} c_i \mathbf{u}_i(t), \quad (3.25)$$

where the cost function landscape is spanned by the coefficients c_i , that need to be optimised over using numerical optimisation methods like those described in Sec. 3.1.3. Generally this basis is made up of trigonometric functions although it could be any basis that spans the space of admissible controls e.g. generalized Chebyshev polynomials. A choice of basis can further be enhanced or modified by a shape function $g(t)$ that fixes the pulse to some initial and/or final value:

$$\mathbf{u}(g(t), t) = \sum_{i=1}^{N_k/2} c_i \frac{\cos \omega_i t}{g(t)} + \sum_{i=N_k/2+1}^{N_k} c_i \frac{\sin \omega_i t}{g(t)}. \quad (3.26)$$

Importantly, the key to expanding the solution space in order to find better pulses using the CRAB approach lies in the randomisation of the frequencies ω_i . During each optimisation process, the ω_i are chosen randomly around the principal harmonics within some interval $[0, \omega_{\max}]$, allowing the pulse shapes to be more diverse and more complex than by simply keeping them fixed at a certain value. The optimisation process can then be parallelised, with several optimisation instances running simultaneously exploring several different sets of random frequencies and the optimal solution can be picked from the final outcomes of all optimisations.

The CRAB approach lends itself very easily to the incorporation of additional features and constraints like the shape function. For example, it is quite easy to start with a trial pulse, say $f(t)$, which cannot be expanded efficiently or exactly in the chosen

basis and to dress it according to

$$u(t) = f(t) \left(1 + \sum_{i=1}^{N_k} c_i u_i(t) \right). \quad (3.27)$$

Another particularly useful alteration to the basic CRAB procedure is what is known as ‘dressed’ CRAB or dCRAB [62], which in a similar vein aims to iteratively re-dress solutions obtained from previous optimisations with new sets of basis functions added onto the existing solution. These *super-iterations* j can be modelled as

$$u^j(t) = c_0^j u^{j-1}(t) + \sum_{i=1}^{N_k} c_i^j u_i^j(t) \quad (3.28)$$

where $u_i^j(t)$ are new basis functions and $u^{j-1}(t)$ is the pulse obtained from a previous, $(j - 1)^{\text{st}}$ optimisation. The coefficient c_0^j can be seen as shifting the solution in the direction of the previous solution pulse while $\{c_i^j\}_{i=1,\dots,N_k}$ move it in new search directions $u_i^j(t)$. This is, in fact, very similar to Powell’s optimisation method which I covered in Sec. 3.1.3.2, wherein a finite set of search directions in cost function space is continuously updated with linear combinations of their optima. dCRAB can be seen as doing the same but with updates sampled from an infinite-dimensional search space. This iterative approach is useful in avoiding local minima and in exploring a far larger search space, avoiding the hard constraint of a finite set of basis functions for each optimisation.

There are several key advantages in the CRAB approach that have led to its widespread use in the QOCT community. For one, the randomization of the control field basis allows for a more comprehensive exploration of the control landscape, which can lead to the discovery of better solutions. It also offers relatively quick convergence as the number of optimisable parameters is usually small when compared to other approaches (such as GRAPE, which we will explore in the next section). Finally, it is very flexible: the basis functions can be altered and constraints can be incorporated quite easily, whether they concern the physical implementation (e.g. the shaping function) or some efficiency concerns (à la dCRAB).

3.3.2 Gradient Ascent Pulse Engineering (GRAPE)

The “Gradient Ascent Pulse Engineering” (GRAPE) algorithm is yet another widely used QOCT numerical method. It was first developed in order to design pulse sequences in NMR spectroscopy [63] and has since been iterated upon and improved a number of times as well as being integrated into several optimal control packages [64–67]. As the name suggests, it is a gradient-based optimisation method and it is used primarily for the preparation of specific target states i.e. the cost function in is the state fidelity given by Eq. (3.21).

The key idea behind GRAPE is to replace continuous control functions, like e.g. those used in the basis functions of CRAB, with piecewise constant control amplitudes $u_j(t_k)$, each applied to the control system at time $t_k \in [0, \tau]$ for a time interval Δt , where τ is the total evolution time. You can view this as discretizing the time-evolution of the system into N_m slices of time $\Delta t = t_{k+1} - t_k$. These slices need not all be of equal size, but for simplicity let us work in the setting where they are, meaning that $\tau = N_m \Delta t$ Reminder: add a figure to illustrate? it's much easier to understand with a figure.

At this point we can pause and notice that since the control amplitudes $u_j(t_k)$ are piecewise constant for all time intervals, they can be treated as a set of parameters that can be optimised using a numerical optimisation algorithm. This gives $N_j \times N_m$ total parameters to optimise, as each j^{th} pulse will be made up of N_m time-steps. Given this relatively large number of parameters, the original GRAPE algorithm included an analysis of how to compute the gradient of the cost function with respect to each $u_j(t_k)$ in order to implement gradient-based optimisation methods like gradient-descent (Eq. (3.15)). Recalling the form of the quantum control Hamiltonian from Eq. (3.18), the propagator for the time-evolution of the quantum system using GRAPE during a single time step Δt at time t_k is

$$U_k(\Delta t) = \exp \left\{ -i\Delta t \left(H_0 + \sum_{j=1}^{N_j} u_j(t_k) \mathcal{O}_{\text{opt}}^{(j)} \right) \right\} \quad (3.29)$$

for some drift component of the Hamiltonian H_0 and some basis of control operators

$\{\mathcal{O}_{\text{opt}}^{(j)}\}_{j=1,\dots,N_j}$. The full evolution of the system can thus be captured by the product of operators (with dependence on Δt removed):

$$U(\tau) = U_{N_m} U_{N_m-1} \dots U_2 U_1, \quad (3.30)$$

such that for some initial state ρ_0 (where we are now working with density matrices rather than state vectors), the final evolved state can be written as

$$\begin{aligned} \rho(\tau) &= U(\tau) \rho_0 U^\dagger(\tau) \\ &= U_{N_m} \dots U_1 \rho_0 U_1^\dagger \dots U_{N_m}^\dagger \end{aligned} \quad (3.31)$$

At this point, in order to derive a way to compute the gradient of the cost function with respect to the parameters, it is necessary to define a cost function. In this case we will use the overlap of the final state $\rho(\tau)$ with respect to some target state ρ_T , a density matrix version of Eq. 3.21:

$$C(\mathbf{u}) = \text{Tr}\left\{\rho_T^\dagger \rho(\tau)\right\}, \quad (3.32)$$

where \mathbf{u} in this case is the set of all $N_j \times N_m$ parameters to be optimised $\mathbf{u} : \{u_j(t_k)\}_{j=1,\dots,N_j}^{k=1,\dots,N_m}$. Using Eq. 3.31 and the cyclic property of the trace we can write

$$\begin{aligned} C(\mathbf{u}) &= \text{Tr}\left\{\rho_T^\dagger U_{N_m} \dots U_1 \rho_0 U_1^\dagger \dots U_{N_m}^\dagger\right\} \\ &= \text{Tr}\left\{U_{k+1}^\dagger \dots U_{N_m}^\dagger \rho_T U_{N_m} \dots U_{k+1} U_k \dots U_1 \rho_0 U_1^\dagger \dots U_j^\dagger\right\} \\ &= \text{Tr}\{\Lambda_k \rho_k\}, \end{aligned} \quad (3.33)$$

where $\Lambda_k = U_{k+1}^\dagger \dots U_{N_m}^\dagger \rho_T U_{N_m} \dots U_{k+1}$ and $\rho_k = U_k \dots U_1 \rho_0 U_1^\dagger \dots U_j^\dagger$.

In order to calculate the gradient of $C(\mathbf{u})$ with respect to each parameter $u_j(t_k)$, we first investigate what happens to U_k when we perturb each parameter by some small amount $\delta u_j(t_k)$. To first order in $\delta u_j(t_k)$ we get

$$\delta U_k = -i \delta u_j(t_k) U_k \int_0^{\Delta t} U_k(t') \mathcal{O}_{\text{opt}}^{(j)} U_k(-t') dt'. \quad (3.34)$$

Chapter 3. Quantum Optimal Control

Then, for small Δt (i.e. when it is much smaller than the norm of the control Hamiltonian), we find that the integral in the expression above can be approximated as the average value of the integrand, leading to:

$$\frac{\delta C(\mathbf{u})}{\delta u_j(t_k)} = - \text{Tr} \left\{ \Lambda_k \left(i\Delta t \left[\mathcal{O}_{\text{opt}}^{(j)}, \rho_k \right] \right) \right\}. \quad (3.35)$$

Using this, it is now possible to implement gradient-based numerical optimisation algorithms in order to find optimal values of \mathbf{u} , in the vein of gradient-descent from Eq. 3.15. The method has been improved upon after the initial algorithm was first published, e.g. in [66] in order to include information about second-derivatives of the cost function, allowing for more complex gradient-based optimisation like quasi-Newton methods (see discussion in Sec. 3.1.3). Recent years have also seen improvements similar to those of dCRAB in the case of the CRAB algorithm, where an iterative optimisation procedure is applied on top of the basic GRAPE algorithm [65]. It would be pertinent to mention, that there are very similar approaches to constructing GRAPE-type pulses out in the literature known as Krotov schemes [68]. The key difference between GRAPE and Krotov is in the update step in the iterative optimisation procedure: where the GRAPE algorithm updates all control parameters in a single iteration at once, the Krotov-based methods do so sequentially. This is a small change, but it leads to completely different convergence behaviour of the algorithm.

Ultimately, GRAPE is a simple and powerful approach to constructing a control pulse, but it can suffer from the large number of parameters that need to be optimised. In more simple settings, where the cost function is smooth and convex, it is a very powerful tool, as on top of the high degree of control over the exact shape of the pulse, it offers a gradient-based method level of convergence. Gradient-based methods, as discussed in Sec. 3.1.3), are efficient and converge rapidly given convexity guarantees, regardless of the number of parameters that describe the control function. However, there is no reason to expect that the cost function landscape will be particularly smooth nor convex in any specific instance, meaning the gradient information obtained in the GRAPE algorithm may not be useful. Furthermore, the gradient evaluation step can

be quite computationally intensive. At the end of the day, one can always construct a GRAPE-type pulse and optimise the many parameters using, for example, a global optimiser like dual-annealing from Sec. 3.1.3.3, but given how high-dimensional the problem might be due to the many parameters involved, this can be a very computationally intensive task.

It is useful to compare GRAPE and CRAB, as each offers a different set of advantages and disadvantages. The effectiveness of CRAB, for example, relies a lot on the choice of basis functions used in constructing the pulse, but the number of parameters to be optimised is generally far lower than that of GRAPE. Both offer a lot of flexibility in terms of incorporating constraints and using different numerical optimisers, although CRAB generally does not include a systematic way to compute cost function gradients, leaving it subject to gradient-free methods.

Part II

Optimising approximate counterdiabatic driving

Chapter 4

Counterdiabatic optimised local driving

A good idea has a way of becoming simpler and solving problems other than that for which it was intended.

Robert Tarjan

In Ch. 2 we established that adiabatic evolution of a quantum system requires very long timescales, without which it experiences non-adiabatic excitations out of its instantaneous eigenstate(s). With the dual motivation of preserving adiabaticity, i.e. enforcing that the system stay in its instantaneous eigenstates, and decreasing the total evolution time in order to avoid decoherence, Sections ?? and ??

This chapter will present a new method, known as Counterdiabatic Optimised Local Driving (COLD), which aims to circumvent the shortcomings of LCD while retaining its advantages of locality and simplicity. It does this by combining ideas from Quantum Optimal Control Theory (QOCT), which were covered extensively in Ch. 3

4.1 Combining counterdiabatic driving and optimal control

The COLD approach begins with the observation that the counterdiabatic schedule will depend on the driving path of the original Hamiltonian $H(\lambda)$ for which it is constructed, where λ is a parameter which captures the time-dependence. This can be seen, for example, in the fact that an exact CD protocol (Eq. (2.40)) implements a drive proportional to the adiabatic gauge potential (AGP) operator \mathcal{A}_λ scaled by the speed of the change in the driving parameter $\dot{\lambda}$. The off-diagonal matrix elements of the AGP operator, responsible for the non-adiabatic effects and given by Eq. (2.32), depend both on the matrix elements of $\partial_\lambda H(\lambda)$ and the instantaneous eigenenergies and eigenstates of $H(\lambda)$. Thus, for a Hamiltonian made up of a set of operators \mathcal{O}_H parameterised by a set of coefficients $\mathbf{h} = \{h_i\}_{i=1,\dots,N_h}$ and the parameter λ , which captures its time-dependence,

The full form of the AGP, given by Eq. (2.32), depends on both the instantaneous eigenstates of $H(\lambda)$ and the matrix

$$H_{\text{COLD}}(\lambda, \boldsymbol{\beta}) = H_0(\lambda) + \boldsymbol{\alpha}(\lambda, \boldsymbol{\beta})\mathcal{O}_{\text{LCD}} + \boldsymbol{\beta}(\lambda)\mathcal{O}_{\text{opt.}} \quad (4.1)$$

4.2 The constraints

4.3 Optimal control toolbox

When it comes to the optimal control component of COLD, it i

4.3.1 COLD-CRAB

Mention differentiability as an advantage! It's a thing.

Also mention how accidentally re-deriving dCRAB.

4.3.2 COLD-GRAPE

Modified version of GRAPE: not the gradient-based approach, but rather one which

Chapter 4. Counterdiabatic optimised local driving

spline, interpolation

Chapter 5

Higher-order countediabatic driving as a cost function

In the previous chapter I presented the idea of combining approximate (local) counter-diabatic driving (LCD) and quantum optimal control in order to improve

5.1 The idea

$$\mathcal{I}_1 = \int_0^\tau dt' \left[\langle \psi_g(t') | \Gamma^2(t') | \psi_g(t') \rangle - (\langle \psi_g(t') | \Gamma(t') | \psi_g(t') \rangle)^2 \right]^{1/2}, \quad (5.1)$$

$$\Gamma(t) = \gamma(t) (\sigma_1^y \sigma_2^x + \sigma_1^x \sigma_2^y), \quad (5.2)$$

$$\mathcal{I}_2 = \int_0^\tau dt' |\gamma(t')|, \quad (5.3)$$

Part III

Applications

Chapter 6

Optimising for state fidelity

6.1 Two-spin annealing

One of the simplest example systems to illustrate the COLD method is that of a two-spin model and a simple annealing protocol:

$$H_0(\lambda) = -2J\sigma_1^z\sigma_2^z - h(\sigma_1^z + \sigma_2^z) + 2h\lambda(\sigma_1^x + \sigma_2^x), \quad (6.1)$$

where the Hamiltonian is parameterised by the set of coefficients $\{J, h, \lambda(t)\}$, with the λ term encoding the time-dependence. For this example we use

$$\lambda(t) = \sin^2\left(\frac{\pi}{2}\sin^2\left(\frac{\pi t}{2\tau}\right)\right), \quad (6.2)$$

such that $\lambda(0) = 0$ and $\lambda(\tau) = 1$. In this way, the transverse field is tuned from 0 to $2h$ as t goes from 0 to τ .

$$\alpha = -\frac{h^2}{4(h\lambda)^2 + h^2 + 4J^2}. \quad (6.3)$$

$$H_\beta(\lambda) = H_0(\lambda) + \sum_{k=1}^{N_k} \beta^k \sin(\pi k \lambda)(\sigma_1^z + \sigma_2^z), \quad (6.4)$$

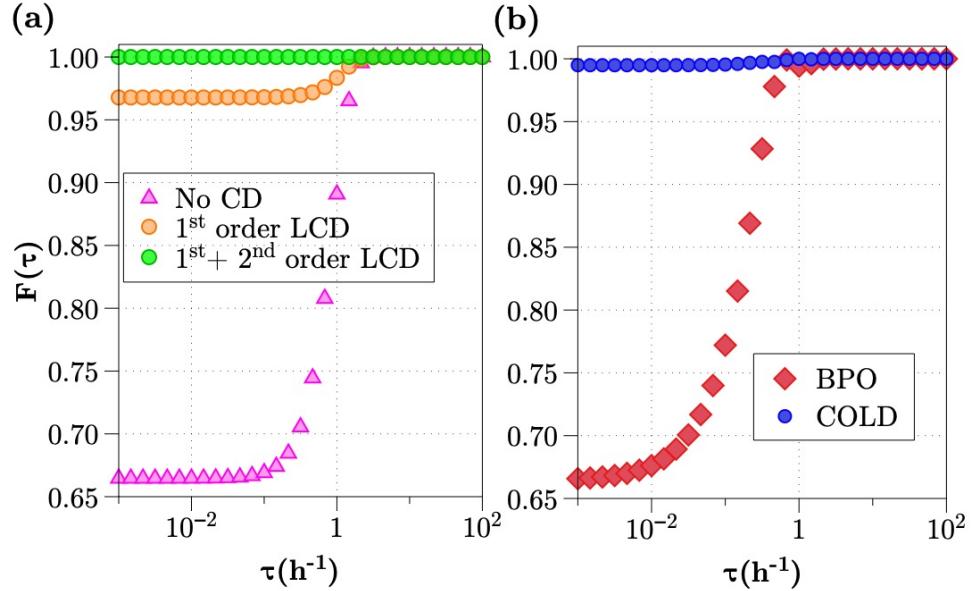


Figure 6.1

6.2 Ising chain

$$H_0(\lambda) = -J \sum_j^{N-1} \sigma_j^z \sigma_{j+1}^z + Z_0 \sum_j^N \sigma_j^z + \lambda X_f \sum_j^N \sigma_j^x, \quad (6.5)$$

$$\mathcal{A}_\lambda^{(1)} = \alpha \sum_j^N \sigma_j^y, \quad (6.6)$$

$$\alpha(\lambda) = \frac{1}{2} \frac{Z_0 X_f}{Z_0^2 + \lambda^2 X_f^2 + 2 J^2}. \quad (6.7)$$

$$\mathcal{A}_\lambda^{(2)} = \alpha \sum_j \sigma_j^y + \gamma \sum_j (\sigma_j^x \sigma_{j+1}^y + \sigma_j^y \sigma_{j+1}^x) + \zeta \sum_j (\sigma_j^z \sigma_{j+1}^y + \sigma_j^y \sigma_{j+1}^z), \quad (6.8)$$

6.3 Transport in a synthetic lattice

$$H_0(t) = - \sum_n J_n(t) (c_n^\dagger c_{n+1} + H.c.) + \sum_n V_n(t) c_n^\dagger c_n, \quad (6.9)$$

$$J_n(t) = J_0(1.1 - \lambda) \quad (6.10)$$

$$V_n(t) = nV_02(\lambda - 1/2), \quad (6.11)$$

$$J_n(t) \rightarrow J_{n,\text{CD}}(t)e^{-i\phi_{n,\text{CD}}(t)}, \quad (6.12)$$

where

$$J_{n,\text{CD}}(t) = \sqrt{J_n(t)^2 + (\alpha_n(t)/\tau)^2}, \quad (6.13)$$

$$\phi_{n,\text{CD}}(t) = \arctan\left(-\frac{J_n(t)\tau}{\alpha_n(t)}\right), \quad (6.14)$$

6.4 Preparing GHZ states in a system of frustrated spins

Multipartite entanglement is a powerful resource for quantum computing, quantum cryptography, quantum teleportation, and quantum metrology, offering unique capabilities for information processing, secure communication, high-precision measurements, and understanding the foundations of quantum mechanics. An example of such highly entangled states is the GHZ state on $N > 1$ spins:

$$|\text{GHZ}\rangle = \frac{1}{\sqrt{2}}(|0\rangle^{\otimes N} + |1\rangle^{\otimes N}). \quad (6.15)$$

GHZ states can be prepared via an annealing schedule on a system of frustrated spins.

$$H_0(\lambda) = -J\left(\sum_j^{N-1} \sigma_j^z \sigma_{j+1}^z + \sum_j^{N-2} \sigma_j^z \sigma_{j+2}^z\right) - h(1 - \lambda)\left(\sum_j^N (\sigma_j^z + \sigma_j^x)\right). \quad (6.16)$$

When J is positive, the ground state of $H_0(1)$ is the GHZ state of Eq. 6.15, whereas when it is negative

While measuring the entanglement of a multipartite system is not quite as simple

Chapter 6. Optimising for state fidelity

as in the bipartite case, there exists a notion of entanglement for a system of three spins: namely, the three-tangle, first introduced in Ref. [69], which

$$\begin{aligned}
 T_3(|\psi\rangle) &= 4|d_1 - 2d_2 + 4d_3|, \\
 d_1 &= c_{000}^2 c_{111}^2 + c_{001}^2 c_{110}^2 + c_{010}^2 c_{101}^2 + c_{011}^2 c_{100}^2, \\
 d_2 &= c_{000} c_{001} c_{110} c_{111} + c_{000} c_{010} c_{101} c_{111} + c_{000} c_{011} c_{100} c_{111} \\
 &\quad + c_{001} c_{010} c_{101} c_{110} + c_{001} c_{011} c_{100} c_{110} + c_{010} c_{011} c_{100} c_{101} \\
 d_3 &= c_{000} c_{110} c_{101} c_{011} + c_{100} c_{010} c_{001} c_{111}
 \end{aligned} \tag{6.17}$$

Not the first time this is attempted with CD. [70]

Chapter 7

Higher order AGP as a cost function

7.1 Return to two-spin annealing

Hey there's lots of cool stuff to see here! Check out Appendix C for more nice plots and things. I might just keep the stuff in Fig. 7.1 for now to keep it focused.

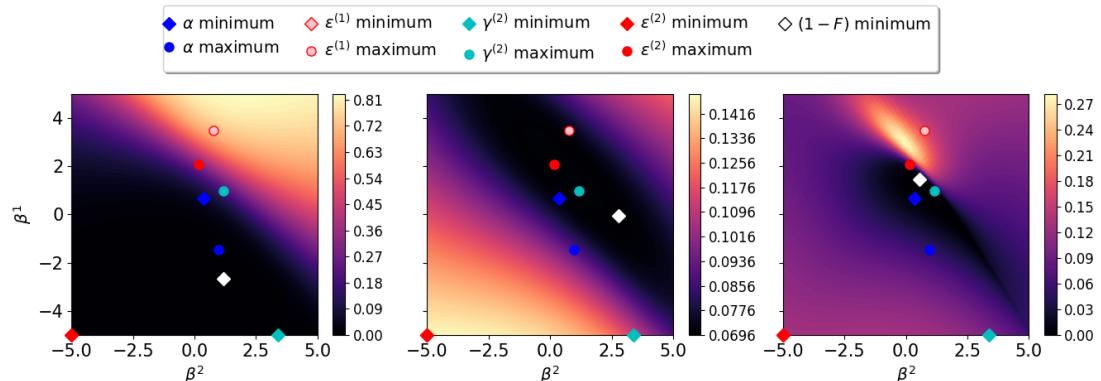


Figure 7.1: Current placeholder for final figure. (a) Only first order CD is applied. (b) only $\sigma^z \sigma^y, \sigma^y \sigma^z$ terms are applied. (c) All second-order terms are applied.

7.2 Random Ising graphs

7.2.1 Maximal Independent Set

7.3 Spin-1/2 XXZ chains of ultracold atoms

Part IV

Conclusion

Chapter 8

Conclusion

Chapter 9

To boldly go

9.1 Exploring the information contained in the AGP

9.2 Applying COLD to useful problems

Appendix A

Rotating spin Hamiltonian

In Chap. 2 I often use the simple example of a spin in a magnetic field, which starts in the $|+\rangle$ state and is rotated from the x direction to the z direction during some total time τ according to the Hamiltonian in Eq. (2.1), which I will reproduce here for convenience:

$$H(\lambda) = -\cos(\lambda)\sigma^x - \sin(\lambda)\sigma^z, \quad (\text{A.1})$$

with $\lambda(t) = \frac{\pi t}{2\tau}$.

Appendix B

Derivation of the CD coefficients for an arbitrary Ising graph

An Ising Hamiltonian for N spins and with both a transverse and longitudinal field can be written as:

$$H(\lambda) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N J_{ij}(\lambda) \sigma_i^z \sigma_j^z + \sum_{i=1}^N \left(X_i(\lambda) \sigma_i^x + Z_i(\lambda) \sigma_i^z \right) \quad (\text{B.1})$$

where the coefficients J_{ij} correspond to couplings between spins i and j . Systems like this can be viewed as undirected graphs, with each spin corresponding to a vertex and each coupling J_{ij} denoting an edge between the corresponding spins. In the case of a weighted graph, the magnitude of each J_{ij} can be viewed as the weight of the corresponding edge. This type of Hamiltonian, for specific values of J_{ij} , X_i and Z_i can be used to describe the two-spin annealing example of Sec. 6.1, the Ising chain from Sec. 6.2 and the frustrated spin model of Sec. ??.

The first order LCD ansatz is just single-spin operators:

$$\mathcal{A}_\lambda^{(1)} = \sum_{i=1}^N \alpha_i(\lambda) \sigma_i^y \quad (\text{B.2})$$

Appendix B. Derivation of the CD coefficients for an arbitrary Ising graph

and the second order can be split up into 4 separate symmetries of operators:

$$\mathcal{A}_\lambda^{(2)} = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \left(\gamma_{ij}(\lambda) \sigma_i^x \sigma_j^y + \bar{\gamma}_{ij}(\lambda) \sigma_i^y \sigma_j^x + \zeta_{ij}(\lambda) \sigma_i^z \sigma_j^y + \bar{\zeta}_{ij}(\lambda) \sigma_i^y \sigma_j^z \right). \quad (\text{B.3})$$

What follows is a derivation that one might call ‘messy’ on a good day. The first order commutators are computed as follows:

$$i[\alpha_i \sigma_i^y, H] = 2\alpha_i \left[\sum_{j=i+1}^N -J_{ij} \left(\sigma_i^x \sigma_j^z + \sigma_i^z \sigma_j^x \right) + X_i \sigma_i^z - Z_i \sigma_i^x \right], \quad (\text{B.4})$$

where I have omitted the dependence on λ of the terms. The second order expansions, sadly, look like this:

$$\begin{aligned} i[\gamma_{ij} \sigma_i^x \sigma_j^y, H(\lambda)] &= 2\gamma_{ij} \left[\sum_{k=1}^{i-1} (J_{ki} \sigma_k^z \sigma_i^y \sigma_j^y - J_{kj} \sigma_k^z \sigma_i^x \sigma_j^x) + \sum_{k=i+1}^{j-1} (J_{ik} \sigma_i^y \sigma_k^z \sigma_j^y - J_{kj} \sigma_i^x \sigma_k^z \sigma_j^x) \right. \\ &\quad \left. + \sum_{k=j+1}^N (J_{ik} \sigma_i^y \sigma_j^y \sigma_k^z - J_{jk} \sigma_i^x \sigma_j^x \sigma_k^z) + Z_i \sigma_i^y \sigma_j^y + X_j \sigma_i^x \sigma_j^z - Z_j \sigma_i^x \sigma_j^x \right] \\ i[\bar{\gamma}_{ij} \sigma_i^y \sigma_j^x, H] &= 2\bar{\gamma}_{ij} \left[\sum_{k=1}^{i-1} (J_{kj} \sigma_k^z \sigma_i^y \sigma_j^y - J_{ki} \sigma_k^z \sigma_i^x \sigma_j^x) + \sum_{k=i+1}^{j-1} (J_{kj} \sigma_i^y \sigma_k^z \sigma_j^y - J_{ik} \sigma_i^x \sigma_k^z \sigma_j^x) \right. \\ &\quad \left. + \sum_{k=j+1}^N (J_{jk} \sigma_i^y \sigma_j^y \sigma_k^z - J_{ik} \sigma_i^x \sigma_j^x \sigma_k^z) + Z_j \sigma_i^y \sigma_j^y + X_i \sigma_i^z \sigma_j^x - Z_i \sigma_i^x \sigma_j^x \right] \\ i[\zeta_{ij} \sigma_i^z \sigma_j^y, H(\lambda)] &= 2\zeta_{ij} \left[- \sum_{k=1}^{i-1} J_{kj} \sigma_k^z \sigma_i^z \sigma_j^y - \sum_{k=i+1}^{j-1} J_{kj} \sigma_i^z \sigma_k^z \sigma_j^y - \sum_{k=j+1}^N J_{jk} \sigma_i^z \sigma_j^x \sigma_k^z \right. \\ &\quad \left. - J_{ij} \sigma_j^x - X_i \sigma_i^y \sigma_j^y + X_j \sigma_i^z \sigma_j^z - Z_j \sigma_i^z \sigma_j^x \right] \\ i[\bar{\zeta}_{ij} \sigma_i^y \sigma_j^z, H(\lambda)] &= 2\bar{\zeta}_{ij} \left[- \sum_{k=1}^{i-1} J_{ki} \sigma_k^z \sigma_i^x \sigma_j^z - \sum_{k=i+1}^{j-1} J_{ik} \sigma_i^x \sigma_k^z \sigma_j^z - \sum_{k=j+1}^N J_{ik} \sigma_i^x \sigma_j^z \sigma_k^z \right. \\ &\quad \left. - J_{ij} \sigma_i^x - X_j \sigma_i^y \sigma_j^y + X_i \sigma_i^z \sigma_j^z - Z_i \sigma_i^x \sigma_j^z \right] \end{aligned} \quad (\text{B.5})$$

Combined, the above commutators along with the coefficients of $\partial_\lambda H$ give the operator $G_\lambda(\mathcal{A}_\lambda^{(1,2)})$ for an ansatz AGP constructed from both single- and two-spin operators (as

Appendix B. Derivation of the CD coefficients for an arbitrary Ising graph

per Eq. (2.43)):

$$\begin{aligned}
G_\lambda(\mathcal{A}_\lambda^{(1,2)}) = & \sum_{i=1}^N \left[(\dot{X}_i - 2\alpha_i Z_i - 2 \sum_{j=1}^{i-1} J_{ji} \zeta_{ji} - 2 \sum_{j=i+1}^N J_{ij} \bar{\zeta}_{ij}) \sigma_i^x \right. \\
& + (\dot{Z}_i + 2\alpha_i X_i) \sigma_i^z \Big] \\
& + \sum_{i=1}^{N-1} \sum_{j=i+1}^N \left[(J_{ij} + 2\zeta_{ij} X_j + 2\bar{\zeta}_{ij} X_i) \sigma_i^z \sigma_j^z \right. \\
& + (2\gamma_{ij} Z_i + 2\bar{\gamma}_{ij} Z_j - 2\zeta_{ij} X_i - 2\bar{\zeta}_{ij} X_j) \sigma_i^y \sigma_j^y \\
& + (2\gamma_{ij} Z_j + 2\bar{\gamma}_{ij} Z_i) \sigma_i^x \sigma_j^x \\
& + (-2\alpha_i J_{ij} + 2\gamma_{ij} X_j - 2\bar{\zeta}_{ij} Z_i) \sigma_i^x \sigma_j^z \\
& + (-2\alpha_j J_{ij} + 2\bar{\gamma}_{ij} X_i - 2\zeta_{ij} Z_j) \sigma_i^z \sigma_j^x \\
& + \sum_{k=1}^{i-1} \left[(2\gamma_{ij} J_{ki} + 2\bar{\gamma}_{ij} J_{kj}) \sigma_k^z \sigma_i^y \sigma_j^y + (2\gamma_{ij} J_{kj} + 2\bar{\gamma}_{ij} J_{ki}) \sigma_k^z \sigma_i^x \sigma_j^x \right. \\
& + (-2\zeta_{ij} J_{kj} - 2\zeta_{kj} J_{ij}) \sigma_k^z \sigma_i^z \sigma_j^x \Big] \\
& + \sum_{k=i+1}^{j-1} \left[(2\gamma_{ij} J_{ik} + 2\bar{\gamma}_{ij} J_{kj}) \sigma_i^y \sigma_k^z \sigma_j^y + (2\gamma_{ij} J_{kj} + 2\bar{\gamma}_{ij} J_{ik}) \sigma_i^z \sigma_j^x \sigma_k^y \right. \\
& + 2\bar{\gamma}_{ij} J_{ik}) \sigma_i^x \sigma_k^z \sigma_j^x + (-2\bar{\zeta}_{ij} J_{ik} - 2\zeta_{ik} J_{ij}) \sigma_k^z \sigma_i^x \sigma_j^z \Big] \\
& + \sum_{k=j+1}^N \left[(2\gamma_{ij} J_{ik} + 2\bar{\gamma}_{ij} J_{jk}) \sigma_i^y \sigma_j^y \sigma_k^z + (2\gamma_{ij} J_{jk} + 2\bar{\gamma}_{ij} J_{ik}) \sigma_i^x \sigma_j^x \sigma_k^z \right. \\
& \left. \left. + (-2\bar{\zeta}_{ij} J_{ik} - 2\bar{\zeta}_{ik} J_{ij}) \sigma_i^z \sigma_j^x \sigma_k^z \right] \right]. \tag{B.6}
\end{aligned}$$

In order to find the coupled set of equations that allow us to compute each of the coefficients in the approximate AGP according to the LCD approach, we need to minimise the action $\mathcal{S} = \text{Tr}[G_\lambda^2]$ with respect to each of the coefficients. As the Pauli operators and their tensor products are traceless, this means that the action is merely the sum of the squares of all the orthogonal operator coefficients of G_λ . Minimising \mathcal{S} with respect

Appendix B. Derivation of the CD coefficients for an arbitrary Ising graph

to each α_i gives:

$$\begin{aligned}
& \alpha_i \left[2Z_i^2 + 2X_i^2 + \sum_{j=1}^{i-1} 2J_{ji}^2 + \sum_{i+1}^N 2J_{ij}^2 \right] \\
& \sum_{j=i+1}^N \gamma_{ij} \left[-2J_{ij}X_j \right] + \sum_{j=1}^{i-1} \bar{\gamma}_{ji} \left[-2J_{ji}X_j \right] \\
& \sum_{j=i+1}^N \bar{\zeta}_{ij} \left[4J_{ij}Z_i \right] + \sum_{j=1}^{i-1} \zeta_{ji} \left[4J_{ji}Z_i \right] \\
& = Z_i \dot{X}_i - X_i \dot{Z}_i,
\end{aligned} \tag{B.7}$$

where i is fixed. Fixing i and j and minimising with respect to each γ_{ij} gives:

$$\begin{aligned}
& \alpha_i \left[-X_j J_{ij} \right] + \zeta_{ij} \left[-X_i Z_i \right] + \bar{\zeta}_{ij} \left[-2X_j Z_i \right] \\
& + \gamma_{ij} \left[Z_i^2 + Z_j^2 + X_j^2 + \sum_{k=1}^{i-1} (J_{ki}^2 + J_{kj}^2) + \sum_{k=i+1}^{j-1} (J_{ik}^2 + J_{kj}^2) + \sum_{k=j+1}^N (J_{ik}^2 + J_{jk}^2) \right] \\
& + \bar{\gamma}_{ij} \left[2Z_i Z_j + \sum_{k=1}^{i-1} 2J_{ki} J_{kj} + \sum_{k=i+1}^{j-1} 2J_{ik} J_{kj} + \sum_{k=j+1}^N 2J_{ik} J_{jk} \right] = 0
\end{aligned} \tag{B.8}$$

and likewise for each $\bar{\gamma}$:

$$\begin{aligned}
& \alpha_j \left[-X_i J_{ij} \right] + \zeta_{ij} \left[-2X_i Z_j \right] + \bar{\zeta}_{ij} \left[-X_j Z_j \right] \\
& + \bar{\gamma}_{ij} \left[Z_i^2 + Z_j^2 + X_i^2 + \sum_{k=1}^{i-1} (J_{ki}^2 + J_{kj}^2) + \sum_{k=i+1}^{j-1} (J_{ik}^2 + J_{kj}^2) + \sum_{k=j+1}^N (J_{ik}^2 + J_{jk}^2) \right] \\
& + \gamma_{ij} \left[2Z_i Z_j + \sum_{k=1}^{i-1} 2J_{ki} J_{kj} + \sum_{k=i+1}^{j-1} 2J_{ik} J_{kj} + \sum_{k=j+1}^N 2J_{ik} J_{jk} \right] = 0.
\end{aligned} \tag{B.9}$$

Appendix B. Derivation of the CD coefficients for an arbitrary Ising graph

Finally, for fixed i, j , we minimise with respect to ζ_{ij} :

$$\begin{aligned}
& \alpha_j [4Z_j J_{ij}] + \gamma_{ij} [-2X_i Z_i] + \bar{\gamma}_{ij} [-4X_i Z_j] \\
& + \zeta_{ij} [2Z_j^2 + 2X_i^2 + 2X_j^2 + \sum_{k=1}^{i-1} 2J_{kj}^2 + \sum_{k=i+1}^{j-1} 2J_{jk}^2] + \bar{\zeta}_{ij} [4X_i X_j] \\
& + \sum_{k=1}^{i-1} \zeta_{kj} [2J_{ij} J_{kj}] + \sum_{k=1}^{j-1} \zeta_{kj} [2J_{ij} J_{kj}] \\
& + \sum_{k=i+1}^{j-1} \bar{\zeta}_{jk} 2J_{ij} J_{jk} + \sum_{k=j+1}^N \bar{\zeta}_{jk} 2J_{ij} J_{jk} = J_{ij} \dot{X}_j - \dot{J}_{ij} X_j
\end{aligned} \tag{B.10}$$

and with respect to $\bar{\zeta}_{ij}$:

$$\begin{aligned}
& \alpha_i [4Z_i J_{ij}] + \gamma_{ij} [-4X_j Z_i] + \bar{\gamma}_{ij} [-2X_j Z_j] \\
& + \bar{\zeta}_{ij} [2Z_i^2 + 2X_i^2 + 2X_j^2 + \sum_{k=i+1}^{j-1} 2J_{ik}^2 + \sum_{k=j+1}^N 2J_{ik}^2] + \zeta_{ij} [4X_i X_j] \\
& + \sum_{k=i+1}^N \bar{\zeta}_{ik} [2J_{ij} J_{ik}] + \sum_{k=j+1}^N \bar{\zeta}_{ik} [2J_{ij} J_{ik}] \\
& + \sum_{k=1}^{i-1} \zeta_{ki} 2J_{ij} J_{ki} + \sum_{k=i+1}^{j-1} \zeta_{ki} 2J_{ij} J_{ki} = J_{ij} \dot{X}_i - \dot{J}_{ij} X_i.
\end{aligned} \tag{B.11}$$

Armed with this knowledge, we can now explore the non-adiabatic effects generated by one- and two-spin operators on any random time-dependent Ising graph Hamiltonian.

Appendix C

More details and plots for the higher order AGP chapter

I have so much stuff to add here, might just leave some here for now.

Appendix C. More details and plots for the higher order AGP chapter

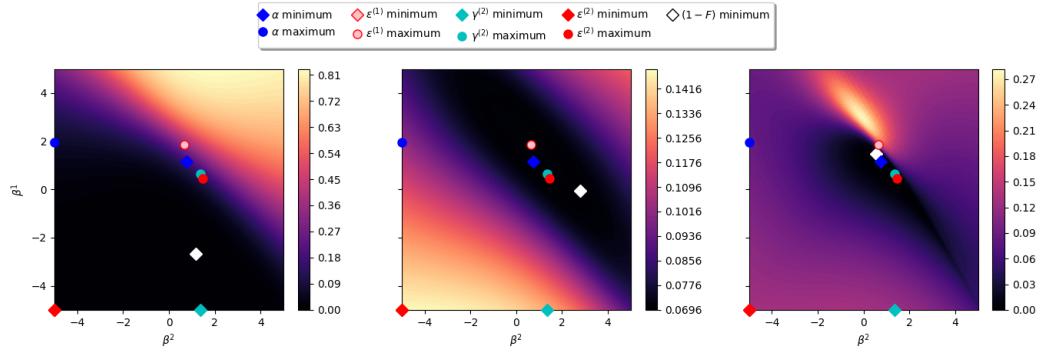


Figure C.1: Placeholder: integral unscaled

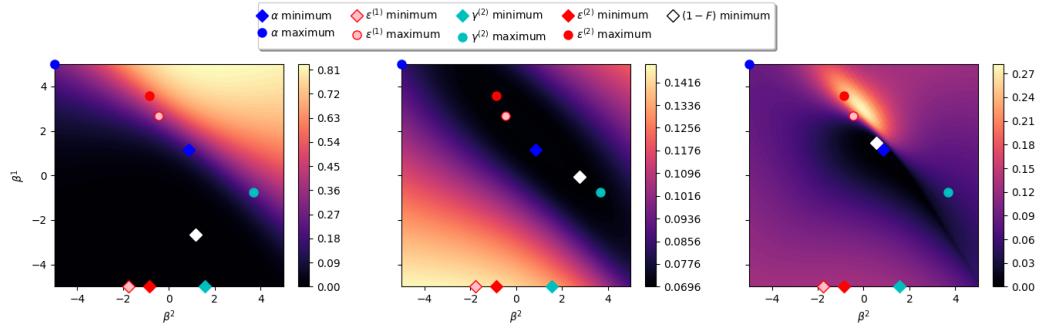


Figure C.2: Placeholder: max amplitude unscaled

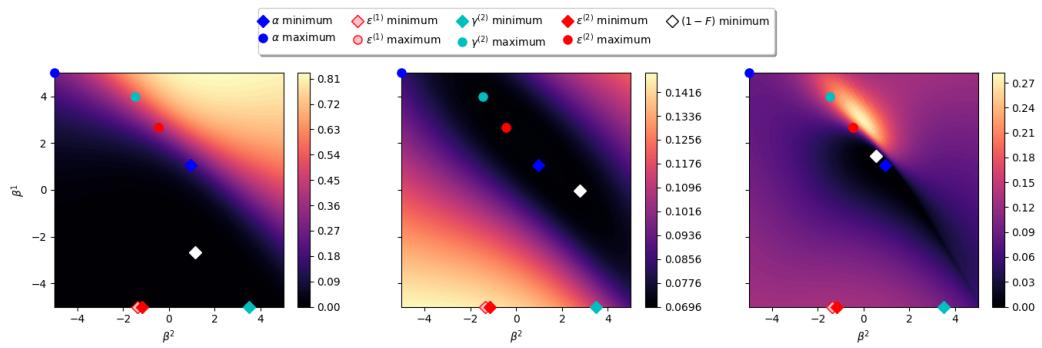


Figure C.3: Placeholder: max amplitude scaled by Hamiltonian norm

Bibliography

- [1] I. Čepaitė, “COLD: Counterdiabatic Optimised Local Driving,” Jan. 2023. [Online]. Available: <https://ievacepaite.com/2023/01/31/cold-counterdiabatic-optimised-local-driving/>
- [2] I. Čepaitė, A. Polkovnikov, A. J. Daley, and C. W. Duncan, “Counterdiabatic Optimized Local Driving,” *PRX Quantum*, vol. 4, no. 1, p. 010312, Jan. 2023, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PRXQuantum.4.010312>
- [3] I. Dimitrova, S. Flannigan, Y. K. Lee, H. Lin, J. Amato-Grill, N. Jepsen, I. Čepaitė, A. J. Daley, and W. Ketterle, “Many-body spin rotation by adiabatic passage in spin-1/2 XXZ chains of ultracold atoms,” *Quantum Science and Technology*, vol. 8, no. 3, p. 035018, May 2023, publisher: IOP Publishing. [Online]. Available: <https://dx.doi.org/10.1088/2058-9565/acd2fb>
- [4] M. Kolodrubetz, D. Sels, P. Mehta, and A. Polkovnikov, “Geometry and non-adiabatic response in quantum and classical systems,” *Physics Reports*, vol. 697, pp. 1–87, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0370157317301989>
- [5] C. Jarzynski, “Geometric Phases and Anholonomy for a Class of Chaotic Classical Systems,” *Physical Review Letters*, vol. 74, no. 10, pp. 1732–1735, Mar. 1995, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.74.1732>

Bibliography

- [6] M. V. Berry, “Transitionless quantum driving,” *Journal of Physics A: Mathematical and Theoretical*, vol. 42, no. 36, p. 365303, Sep. 2009. [Online]. Available: <https://iopscience.iop.org/article/10.1088/1751-8113/42/36/365303>
- [7] M. Demirplak and S. A. Rice, “Adiabatic Population Transfer with Control Fields,” *The Journal of Physical Chemistry A*, vol. 107, no. 46, pp. 9937–9945, Nov. 2003, publisher: American Chemical Society. [Online]. Available: <https://doi.org/10.1021/jp030708a>
- [8] D. Guéry-Odelin, A. Ruschhaupt, A. Kiely, E. Torrontegui, S. Martínez-Garaot, and J. Muga, “Shortcuts to adiabaticity: Concepts, methods, and applications,” *Reviews of Modern Physics*, vol. 91, no. 4, p. 045001, Oct. 2019, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/RevModPhys.91.045001>
- [9] S. Pancharatnam, “Generalized theory of interference, and its applications,” *Proceedings of the Indian Academy of Sciences - Section A*, vol. 44, no. 5, pp. 247–262, Nov. 1956. [Online]. Available: <https://doi.org/10.1007/BF03046050>
- [10] H. C. Longuet-Higgins, U. Öpik, M. H. L. Pryce, and R. A. Sack, “Studies of the Jahn-Teller Effect. II. The Dynamical Problem,” *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 244, no. 1236, pp. 1–16, 1958, publisher: The Royal Society. [Online]. Available: <https://www.jstor.org/stable/100248>
- [11] M. V. Berry, “Quantal Phase Factors Accompanying Adiabatic Changes,” *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 392, no. 1802, pp. 45–57, 1984, publisher: The Royal Society. [Online]. Available: <https://www.jstor.org/stable/2397741>
- [12] M. Nakahara, *Geometry, Topology and Physics, Second Edition*. Taylor & Francis, Jun. 2003. [Online]. Available: <http://www.crcnetbase.com/doi/book/10.1201/9781420056945>

Bibliography

- [13] C. S. Kahane, “Generalizations of the Riemann-Lebesgue and Cantor-Lebesgue lemmas,” *Czechoslovak Mathematical Journal*, vol. 30, no. 1, pp. 108–117, 1980, publisher: Institute of Mathematics, Academy of Sciences of the Czech Republic. [Online]. Available: <https://eudml.org/doc/13181>
- [14] B. W. Reichardt, “The quantum adiabatic optimization algorithm and local minima,” in *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, ser. STOC ’04. New York, NY, USA: Association for Computing Machinery, Jun. 2004, pp. 502–510. [Online]. Available: <https://dl.acm.org/doi/10.1145/1007352.1007428>
- [15] A. Childs, “LECTURE 18: The quantum adiabatic theorem,” 2008. [Online]. Available: <http://www.cs.umd.edu/~amchilds/teaching/w08/l18.pdf>
- [16] D. Sels and A. Polkovnikov, “Minimizing irreversible losses in quantum systems by local counterdiabatic driving,” *Proceedings of the National Academy of Sciences*, vol. 114, no. 20, pp. E3909–E3916, May 2017, publisher: National Academy of Sciences Section: PNAS Plus. [Online]. Available: <https://www.pnas.org/content/114/20/E3909>
- [17] P. W. Claeys, M. Pandey, D. Sels, and A. Polkovnikov, “Floquet-engineering counterdiabatic protocols in quantum many-body systems,” *Physical Review Letters*, vol. 123, no. 9, p. 090602, Aug. 2019, arXiv: 1904.03209. [Online]. Available: <http://arxiv.org/abs/1904.03209>
- [18] M. Pandey, P. W. Claeys, D. K. Campbell, A. Polkovnikov, and D. Sels, “Adiabatic Eigenstate Deformations as a Sensitive Probe for Quantum Chaos,” *Physical Review X*, vol. 10, no. 4, p. 041017, Oct. 2020, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevX.10.041017>
- [19] D. K. Nandy, T. Cadez, B. Dietz, A. Andreanov, and D. Rosa, “Delayed Thermalization in Mass-Deformed SYK,” *Physical Review B*, vol. 106, no. 24, p.

Bibliography

- 245147, Dec. 2022, arXiv:2206.08599 [cond-mat, physics:hep-th, physics:quant-ph]. [Online]. Available: <http://arxiv.org/abs/2206.08599>
- [20] F. Petziol, “Accelerated adiabatic quantum control,” Doctoral thesis, Università di Parma. Dipartimento di Scienze Matematiche, fisiche e informatiche, Mar. 2020, accepted: 2020-04-18T07:16:03Z Journal Abbreviation: Controllo adiabatico quantistico accelerato. [Online]. Available: <https://www.repository.unipr.it/handle/1889/4010>
- [21] S. Ebadi, A. Keesling, M. Cain, T. T. Wang, H. Levine, D. Bluvstein, G. Semeghini, A. Omran, J.-G. Liu, R. Samajdar, X.-Z. Luo, B. Nash, X. Gao, B. Barak, E. Farhi, S. Sachdev, N. Gemelke, L. Zhou, S. Choi, H. Pichler, S.-T. Wang, M. Greiner, V. Vuletić, and M. D. Lukin, “Quantum optimization of maximum independent set using Rydberg atom arrays,” *Science*, vol. 376, no. 6598, pp. 1209–1215, Jun. 2022, publisher: American Association for the Advancement of Science. [Online]. Available: <https://www.science.org/doi/10.1126/science.abo6587>
- [22] H. Pichler, S.-T. Wang, L. Zhou, S. Choi, and M. D. Lukin, “Quantum Optimization for Maximum Independent Set Using Rydberg Atom Arrays,” *arXiv:1808.10816 [cond-mat, physics:physics, physics:quant-ph]*, Aug. 2018, arXiv: 1808.10816. [Online]. Available: <http://arxiv.org/abs/1808.10816>
- [23] G. Pelegrí, A. J. Daley, and J. D. Pritchard, “High-fidelity multiqubit Rydberg gates via two-photon adiabatic rapid passage,” *Quantum Science and Technology*, vol. 7, no. 4, p. 045020, Aug. 2022, publisher: IOP Publishing. [Online]. Available: <https://dx.doi.org/10.1088/2058-9565/ac823a>
- [24] L. D’Alessio, Y. Kafri, A. Polkovnikov, and M. Rigol, “From quantum chaos and eigenstate thermalization to statistical mechanics and thermodynamics,” *Advances in Physics*, vol. 65, no. 3, pp. 239–362, May 2016, publisher: Taylor & Francis .eprint: <https://doi.org/10.1080/00018732.2016.1198134>. [Online]. Available: <https://doi.org/10.1080/00018732.2016.1198134>

Bibliography

- [25] C. Jarzynski, “Generating shortcuts to adiabaticity in quantum and classical dynamics,” *Physical Review A*, vol. 88, no. 4, p. 040101, Oct. 2013, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.88.040101>
- [26] N. Goldman and J. Dalibard, “Periodically Driven Quantum Systems: Effective Hamiltonians and Engineered Gauge Fields,” *Physical Review X*, vol. 4, no. 3, p. 031027, Aug. 2014, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevX.4.031027>
- [27] H. Ball, M. J. Biercuk, A. R. R. Carvalho, J. Chen, M. Hush, L. A. D. Castro, L. Li, P. J. Liebermann, H. J. Slatyer, C. Edmunds, V. Frey, C. Hempel, and A. Milne, “Software tools for quantum control: improving quantum computer performance through noise and error suppression,” *Quantum Science and Technology*, vol. 6, no. 4, p. 044011, Sep. 2021, publisher: IOP Publishing. [Online]. Available: <https://dx.doi.org/10.1088/2058-9565/abdca6>
- [28] A. Omran, H. Levine, A. Keesling, G. Semeghini, T. T. Wang, S. Ebadi, H. Bernien, A. S. Zibrov, H. Pichler, S. Choi, J. Cui, M. Rossignolo, P. Rembold, S. Montangero, T. Calarco, M. Endres, M. Greiner, V. Vuletić, and M. D. Lukin, “Generation and manipulation of Schrödinger cat states in Rydberg atom arrays,” *Science*, vol. 365, no. 6453, pp. 570–574, Aug. 2019, publisher: American Association for the Advancement of Science. [Online]. Available: <https://www.science.org/doi/10.1126/science.aax9743>
- [29] N. H. Le, M. Cykiert, and E. Ginossar, “Robust optimal control of interacting multi-qubit systems for quantum sensing,” Oct. 2021, arXiv:2110.12560 [quant-ph]. [Online]. Available: <http://arxiv.org/abs/2110.12560>
- [30] C. P. Koch, U. Boscain, T. Calarco, G. Dirr, S. Filipp, S. J. Glaser, R. Kosloff, S. Montangero, T. Schulte-Herbrüggen, D. Sugny, and F. K. Wilhelm, “Quantum optimal control in quantum technologies. Strategic report on current status, visions and goals for research in Europe,” *EPJ Quantum Technology*, vol. 9, no. 1,

Bibliography

- pp. 1–60, Dec. 2022, number: 1 Publisher: SpringerOpen. [Online]. Available: <https://epjquantumtechnology.springeropen.com/articles/10.1140/epjqt/s4050-022-00138-x>
- [31] S. J. Glaser, U. Boscain, T. Calarco, C. P. Koch, W. Köckenberger, R. Kosloff, I. Kuprov, B. Luy, S. Schirmer, T. Schulte-Herbrüggen, D. Sugny, and F. K. Wilhelm, “Training Schrödinger’s cat: quantum optimal control,” *The European Physical Journal D*, vol. 69, no. 12, p. 279, Dec. 2015. [Online]. Available: <https://doi.org/10.1140/epjd/e2015-60464-1>
- [32] X.-M. Zhang, Z. Wei, R. Asad, X.-C. Yang, and X. Wang, “When does reinforcement learning stand out in quantum control? A comparative study on state preparation,” *npj Quantum Information*, vol. 5, no. 1, pp. 1–7, Oct. 2019, number: 1 Publisher: Nature Publishing Group. [Online]. Available: <https://www.nature.com/articles/s41534-019-0201-8>
- [33] P. Rooney, A. Bloch, and C. Rangan, “Decoherence Control and Purification of Two-dimensional Quantum Density Matrices under Lindblad Dissipation,” Jan. 2012, arXiv:1201.0399 [math-ph, physics:quant-ph]. [Online]. Available: <http://arxiv.org/abs/1201.0399>
- [34] D. D’Alessandro, *Introduction to Quantum Control and Dynamics*, 2nd ed. New York: Chapman and Hall/CRC, Jul. 2021.
- [35] A. Wald, *Statistical decision functions*, ser. Statistical decision functions. Oxford, England: Wiley, 1950, pages: ix, 179.
- [36] G. Dirr and U. Helmke, “Lie Theory for Quantum Control,” *GAMM-Mitteilungen*, vol. 31, no. 1, pp. 59–93, 2008, eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/gamm.200890003>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/gamm.200890003>
- [37] W. Fleming and R. Rishel, “The Optimal Control Problem,” in *Deterministic and Stochastic Optimal Control*, ser. Applications of Mathematics, W. Fleming and

Bibliography

- R. Rishel, Eds. New York, NY: Springer, 1975, pp. 20–59. [Online]. Available: https://doi.org/10.1007/978-1-4612-6380-7_2
- [38] F. vom Ende, “Reachability in Controlled Markovian Quantum Systems,” Ph.D. dissertation, Technische Universität München, 2020. [Online]. Available: <https://mediatum.ub.tum.de/1559809>
- [39] O. L. Mangasarian, “Sufficient Conditions for the Optimal Control of Nonlinear Systems,” *SIAM Journal on Control*, vol. 4, no. 1, pp. 139–152, Feb. 1966, publisher: Society for Industrial and Applied Mathematics. [Online]. Available: <https://pubs.siam.org/doi/10.1137/0304013>
- [40] V. Boltyanski, H. Martini, and V. Soltan, “Nonclassical Variational Calculus,” in *Geometric Methods and Optimization Problems*, ser. Combinatorial Optimization, V. Boltyanski, H. Martini, and V. Soltan, Eds. Boston, MA: Springer US, 1999, pp. 1–230. [Online]. Available: https://doi.org/10.1007/978-1-4615-5319-9_1
- [41] J. Yong and X. Y. Zhou, “Dynamic Programming and HJB Equations,” in *Stochastic Controls: Hamiltonian Systems and HJB Equations*, ser. Applications of Mathematics, J. Yong and X. Y. Zhou, Eds. New York, NY: Springer, 1999, pp. 157–215. [Online]. Available: https://doi.org/10.1007/978-1-4612-1466-3_4
- [42] M. Sniedovich, *Dynamic Programming: Foundations and Principles, Second Edition*, 2nd ed. Boca Raton: CRC Press, Sep. 2010.
- [43] E. Süli and D. F. Mayers, *An Introduction to Numerical Analysis*. Cambridge: Cambridge University Press, 2003. [Online]. Available: <https://www.cambridge.org/core/books/an-introduction-to-numerical-analysis/FD8BCAD7FE68002E2179DFF68B8B7237>
- [44] L. Bottou and O. Bousquet, “The Tradeoffs of Large Scale Learning,” in *Advances in Neural Information Processing Systems*, vol. 20. Curran Associates, Inc., 2007. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2007/0d3180d672e08b4c5312dcdafdf6ef36-Abstract.html

Bibliography

- [45] M. R. Bonyadi and Z. Michalewicz, “Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review,” *Evolutionary Computation*, vol. 25, no. 1, pp. 1–54, Mar. 2017. [Online]. Available: https://doi.org/10.1162/EVCO_r_00180
- [46] P. A. Vikhar, “Evolutionary algorithms: A critical review and its future prospects,” in *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, Dec. 2016, pp. 261–265.
- [47] C. Tsallis and D. A. Stariolo, “Generalized simulated annealing,” *Physica A: Statistical Mechanics and its Applications*, vol. 233, no. 1, pp. 395–406, Nov. 1996. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0378437196002713>
- [48] J. A. Nelder and R. Mead, “A Simplex Method for Function Minimization,” *The Computer Journal*, vol. 7, no. 4, pp. 308–313, Jan. 1965. [Online]. Available: <https://doi.org/10.1093/comjnl/7.4.308>
- [49] M. J. D. Powell, “An efficient method for finding the minimum of a function of several variables without calculating derivatives,” *The Computer Journal*, vol. 7, no. 2, pp. 155–162, Jan. 1964. [Online]. Available: <https://doi.org/10.1093/comjnl/7.2.155>
- [50] R. P. Brent, *Algorithms for Minimization Without Derivatives*. Courier Corporation, Jan. 2002.
- [51] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by Simulated Annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, May 1983, publisher: American Association for the Advancement of Science. [Online]. Available: <https://www.science.org/doi/10.1126/science.220.4598.671>
- [52] S. Chib and E. Greenberg, “Understanding the Metropolis-Hastings Algorithm,” *The American Statistician*, vol. 49, no. 4, pp. 327–335, 1995, publisher: [American Statistical Association, Taylor & Francis, Ltd.]. [Online]. Available: <https://www.jstor.org/stable/2684568>

Bibliography

- [53] S. G. Schirmer, H. Fu, and A. I. Solomon, “Complete controllability of quantum systems,” *Physical Review A*, vol. 63, no. 6, p. 063410, May 2001. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.63.063410>
- [54] N. Khaneja, R. Brockett, and S. J. Glaser, “Time optimal control in spin systems,” *Physical Review A*, vol. 63, no. 3, p. 032308, Feb. 2001, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.63.032308>
- [55] P. Doria, T. Calarco, and S. Montangero, “Optimal Control Technique for Many-Body Quantum Dynamics,” *Physical Review Letters*, vol. 106, no. 19, p. 190501, May 2011, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.106.190501>
- [56] T. Caneva, T. Calarco, and S. Montangero, “Chopped random-basis quantum optimization,” *Physical Review A*, vol. 84, no. 2, p. 022326, Aug. 2011, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.84.022326>
- [57] S. R. White, “Density matrix formulation for quantum renormalization groups,” *Physical Review Letters*, vol. 69, no. 19, pp. 2863–2866, Nov. 1992, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.69.2863>
- [58] U. Schollwöck, “The density-matrix renormalization group,” *Reviews of Modern Physics*, vol. 77, no. 1, pp. 259–315, Apr. 2005, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/RevModPhys.77.259>
- [59] ——, “The density-matrix renormalization group in the age of matrix product states,” *Annals of Physics*, vol. 326, no. 1, pp. 96–192, Jan. 2011. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0003491610001752>

Bibliography

- [60] C. Brif, R. Chakrabarti, and H. Rabitz, “Control of quantum phenomena: past, present and future,” *New Journal of Physics*, vol. 12, no. 7, p. 075008, Jul. 2010, publisher: IOP Publishing. [Online]. Available: <https://dx.doi.org/10.1088/1367-2630/12/7/075008>
- [61] J. H. M. Jensen, F. S. Møller, J. J. Sørensen, and J. F. Sherson, “Approximate dynamics leading to more optimal control: Efficient exact derivatives,” *Physical Review A*, vol. 103, no. 6, p. 062612, Jun. 2021, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.103.062612>
- [62] N. Rach, M. M. Müller, T. Calarco, and S. Montangero, “Dressing the chopped-random-basis optimization: A bandwidth-limited access to the trap-free landscape,” *Physical Review A*, vol. 92, no. 6, p. 062343, Dec. 2015, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.92.062343>
- [63] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, “Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms,” *Journal of Magnetic Resonance*, vol. 172, no. 2, pp. 296–305, Feb. 2005. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1090780704003696>
- [64] P. de Fouquieres, S. G. Schirmer, S. J. Glaser, and I. Kuprov, “Second order gradient ascent pulse engineering,” *Journal of Magnetic Resonance*, vol. 212, no. 2, pp. 412–417, Oct. 2011, arXiv:1102.4096 [math-ph, physics:quant-ph]. [Online]. Available: <http://arxiv.org/abs/1102.4096>
- [65] Y. Chen, Y. Hao, Z. Wu, B.-Y. Wang, R. Liu, Y. Hou, J. Cui, M.-H. Yung, and X. Peng, “Iterative Gradient Ascent Pulse Engineering algorithm for quantum optimal control,” Dec. 2022, arXiv:2212.02806 [quant-ph]. [Online]. Available: <http://arxiv.org/abs/2212.02806>

Bibliography

- [66] S. Machnes, U. Sander, S. J. Glaser, P. De Fouquières, A. Gruslys, S. Schirmer, and T. Schulte-Herbrüggen, “Comparing, optimizing, and benchmarking quantum-control algorithms in a unifying programming framework,” *Physical Review A*, vol. 84, no. 2, p. 022305, Aug. 2011. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.84.022305>
- [67] J. R. Johansson, P. D. Nation, and F. Nori, “QuTiP 2: A Python framework for the dynamics of open quantum systems,” *Computer Physics Communications*, vol. 184, no. 4, pp. 1234–1240, Apr. 2013. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0010465512003955>
- [68] D. M. Reich, M. Ndong, and C. P. Koch, “Monotonically convergent optimization in quantum control using Krotov’s method,” *The Journal of Chemical Physics*, vol. 136, no. 10, p. 104103, Mar. 2012. [Online]. Available: <https://doi.org/10.1063/1.3691827>
- [69] V. Coffman, J. Kundu, and W. K. Wootters, “Distributed entanglement,” *Physical Review A*, vol. 61, no. 5, p. 052306, Apr. 2000, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.61.052306>
- [70] D. Sun, P. Chandarana, Z.-H. Xin, and X. Chen, “Optimizing counterdiabaticity by variational quantum circuits,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 380, no. 2239, p. 20210282, Nov. 2022, publisher: Royal Society. [Online]. Available: <https://royalsocietypublishing.org/doi/full/10.1098/rsta.2021.0282>

Bibliography

Bibliography