

Counterdiabatic, Better, Faster, Stronger: Optimal control for approximate counterdiabatic driving

PhD Thesis

Ieva Čepaitė

Quantum Optics and Quantum Many-Body Physics

Department of Physics

University of Strathclyde, Glasgow

June 28, 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

Adiabatic protocols are employed across a variety of quantum technologies, from implementing state preparation and individual operations that are building blocks of larger devices, to higher-level protocols in quantum annealing and adiabatic quantum computation. The problem of speeding up these processes has garnered a large amount of interest, resulting in a menagerie of approaches, most notably quantum optimal control and shortcuts to adiabaticity. The two approaches are complementary: optimal control manipulates control fields to steer the dynamics in the minimum allowed time, while shortcuts to adiabaticity aims to retain the adiabatic condition upon speed-up. We outline a new method that combines the two methodologies and takes advantage of the strengths of each. The new technique improves upon approximate local counterdiabatic driving with the addition of time-dependent control fields. We refer to this new method as counterdiabatic optimized local driving (COLD) and we show that it can result in a substantial improvement when applied to annealing protocols, state preparation schemes, entanglement generation, and population transfer on a lattice. We also demonstrate a new approach to the optimization of control fields that does not require access to the wave function or the computation of system dynamics. COLD can be enhanced with existing advanced optimal control methods and we explore this using the chopped randomized basis method and gradient ascent pulse engineering.

Contents

Abstract	ii
Lay Summary	vi
Acknowledgements	vii
Acronyms and abbreviations	viii
List of Figures	viii
1 Introduction	2
1.1 Thesis overview	3
1.2 Publications and manuscripts	4
1.3 Talks and presentations	4
I Background	6
2 Quantum Adiabaticity	7
2.1 The quantum adiabatic theorem	8
2.1.1 Proof of the adiabatic theorem	9
2.1.2 The adiabatic condition: how slow is <i>slow</i> ?	13
2.2 The adiabatic gauge potential	16
2.2.1 The moving frame Hamiltonian	16
2.2.2 Matrix elements of the AGP	18
2.3 Counterdiabatic Driving	20

Contents

2.3.1	Brief interlude: the waiter and the glass of water	23
2.4	The approximate counterdiabatic drive	25
2.4.1	Local counterdiabatic driving	25
2.4.2	Nested commutator expansion	29
3	Quantum Optimal Control	32
3.1	The structure of optimal control problems	33
3.1.1	Mathematical structure	34
3.1.2	Analytic optimisation	37
3.1.3	Numerical optimisation	38
3.2	Quantum optimal control	47
3.3	Quantum optimal control methods	51
3.3.1	Chopped random-basis quantum optimization (CRAB)	51
3.3.2	Gradient Ascent Pulse Engineering (GRAPE)	53
II	Optimising approximate counterdiabatic driving	58
4	Counterdiabatic optimised local driving	59
4.1	Counterdiabatic driving and optimal control	60
4.1.1	The method	64
4.2	Optimal control toolbox	66
5	Adiabatic gauge potential as a cost function	69
5.1	Motivation	70
5.2	Designing a cost function around the counterdiabatic pulse	71
III	Applications of COLD	74
6	Optimising for properties of the state	75
6.1	Two-spin annealing	76
6.2	Ising chain	82
6.2.1	Restricting the driving amplitudes	86

Contents

– DRAFT – June 28, 2023 –

6.3	Transport in a synthetic lattice	87
6.4	Preparing GHZ states in a system of frustrated spins	92
6.4.1	Tripartite GHZ entanglement	95
7	Higher order AGP as a cost function	99
7.1	Return to two-spin annealing	100
7.2	Return to the Ising spin chain	105
7.3	GHZ states and frustrated spins	109
IV	Conclusion	112
8	Summary	113
9	To boldly go	116
9.1	Physical implementation	117
9.2	Extensions of COLD	118
A	Rotating spin Hamiltonian	120
B	Pontryagin maximum principle	123
C	Derivation of the CD coefficients for an arbitrary Ising graph	125
D	Additional plots for the Ising spin chain example	130
E	More details and plots for the higher order AGP chapter	132
	Bibliography	132

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
FO	First order
GRAPE	Gradient Ascent Pulse Engineering
GSA	Generalized Simulated Annealing
LCD	Local Counterdiabatic Driving
PMP	Pontryagin Maximum Principle
QOCT	Quantum Optimal Control Theory
SO	Second order
STA	Shortcuts to Adiabaticity

List of Figures

1	DALL·E turtle	vi
1.1	Thesis outline.	3
2.1	Rotating spin Bloch sphere illustrations	9
2.2	Rotating spin Bloch sphere illustration with counterdiabatic driving	28
3.1	Illustration of optimal control problem structure	35
3.2	Visualising the Nelder-Mead optimisation algorithm.	42
3.3	Visualising optimisers in action	45
3.4	Schematic diagram of open-loop quantum optimal control	49
4.1	A diagrammatic illustration of the COLD method	65
6.1	COLD applied to two-spin annealing	79
6.2	Applying COLD and COLD-CRAB to the Ising chain for 5 spins without constraints on the driving amplitudes.	83
6.3	Applying COLD and COLD-CRAB to the Ising chain for 5 spins with constrained driving amplitudes.	86
6.4	COLD plots for ARP transport in a synthetic lattice	90
6.5	Preparation of GHZ states in a system of frustrated spins.	94
6.6	Preparing 3-spin GHZ states using the 3-tangle as a metric.	97
7.1	Two-spin annealing plots of fidelity versus driving time when minimising different orders of LCD.	101

List of Figures

7.2	Two-spin annealing contour plots for final state fidelity and AGP cost function values.	104
7.3	Plot of final state fidelity for the Ising spin chain for different cost functions and no counterdiabatic component in the implementation.	106
7.4	Plot of final state fidelity for the Ising spin chain for different cost functions with LCD applied.	108
7.5	Contour plots of cost function landscapes for GHZ state preparation in frustrated spin systems (integral cost function).	111
D.1	Plots of how final state fidelities scale using COLD and BPO for different system sizes and optimisable parameters.	131
D.2	Plots of maximum amplitudes of LCD drives for the Ising spin chain. . .	131
E.1	Contour plots of cost function landscapes for GHZ state preparation in frustrated spin systems (maximum amplitude cost function).	133
E.2	Contour plots of cost function landscapes for GHZ state preparation in frustrated spin systems (maximum amplitude cost function) using a bare optimisation pulse.	134
E.3	Contour plots of cost function landscapes for GHZ state preparation in frustrated spin systems (maximum amplitude cost function) using a bare optimisation pulse.	135

Chapter 1

Introduction

A 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 gaining insight into the physics governing such systems or in order to solve complex computational problems. Despite the fact that quantum mechanics has been established for around a century, we have only recently begun to harness the unique features found in the quantum domain, spurned by and further proliferating the rapid progress of highly-controllable quantum systems. We are on the cusp of a new age of quantum technologies, driven by the controlled exploitation of phenomena like entanglement or contextuality and enabling high-precision sensing and quantum simulators, which can allow us to probe and predict the behaviour of quantum systems in ways that could never be done before on paper or using classical computational means. **Reminder: LOTS OF CITATIONS.**

With this development in experimental capabilities, the demand for efficient control protocols and theoretical techniques for the manipulation of quantum systems and their dynamics has increased dramatically. On the one hand, we have experienced an explosive rise in gate-based, digital quantum algorithms, which provide standardized recipes for quantum state manipulation given universal control. On the other, we have witnessed a growth of the field of quantum optimal control theory, which allows one to

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

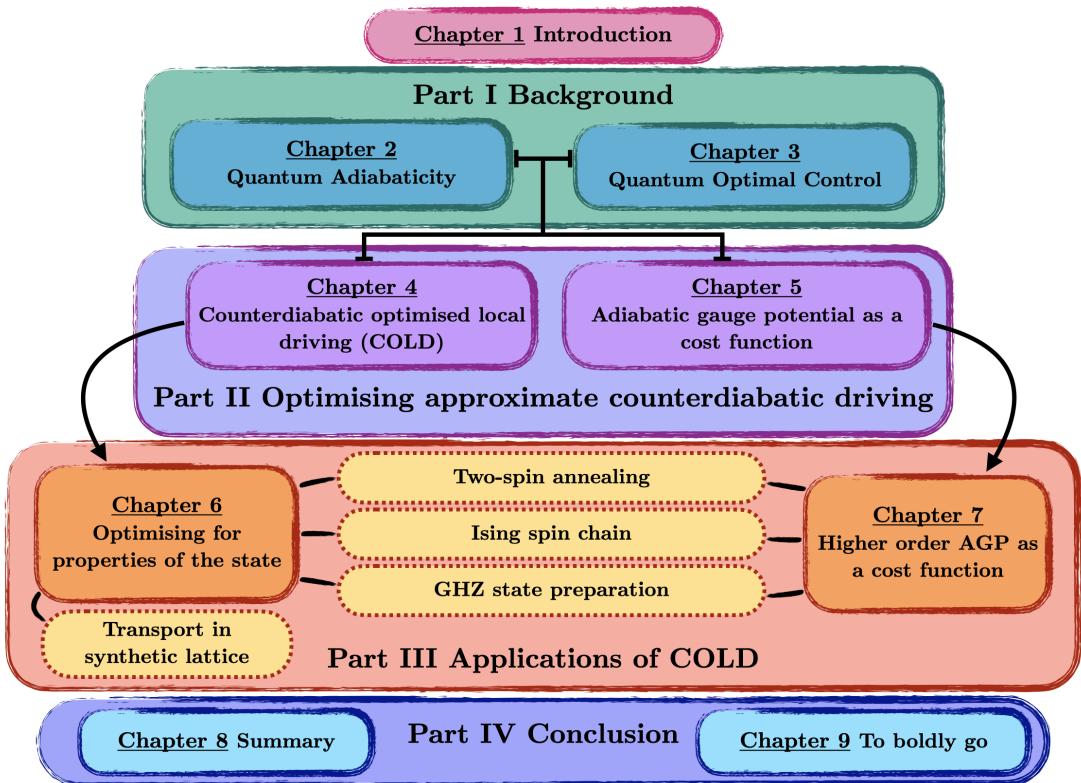


Figure 1.1: Thesis outline. In Part I, we will introduce the concept of quantum adiabaticity, the counterdiabatic driving (CD) method and its approximations as well as quantum optimal control theory (QOCT) and several optimisation techniques that we will apply later in the thesis. Then, in Part II, we will combine ideas from CD and QOCT in order to develop a new method for speeding up adiabatic dynamics and the focal point of this thesis: “Counterdiabatic Optimised Local Driving” or COLD. We will then extend the method with a new optimisation metric based on information about non-adiabatic effects experienced by the system in fast driving. In Part III we will numerically implement the COLD method and its extensions in several different quantum systems to evaluate their performance and compare it to existing techniques. Then, finally, in Part IV, we will conclude with a summary of the thesis and a look towards the future and several open questions that arise from the work presented here.

1.1 Thesis overview

The structure of this thesis is as follows: in Ch. 2 then in Ch. 3. And THEN in Ch. 4 followed by Ch. 5. Finally, in Ch. 6 and Ch. 7. Then a summary in Ch. 8 and future outlook in Ch. 9.

1.2 Publications and manuscripts

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

- (1) **Counterdiabatic Optimised Local Driving**, *Ieva Čepaitė, 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)

Chapter 1. Introduction

- “*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. In classical thermodynamics, an adiabatic process is one where no heat is transferred between a system and its environment. On a microscopic quantum mechanical level, this means not changing the occupation/population of Hamiltonian eigenstates. The quantum adiabatic theorem then describes how slowly changes to the Hamiltonian and therefore the eigenstates have to be made so as not to change the distribution. 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 we will derive the adiabatic condition and explore what happens when the rate of change in the

Chapter 2. Quantum Adiabaticity

Hamiltonian parameters is too fast for adiabaticity. As we will find, the non-adiabatic effects that 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 in a chosen basis 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

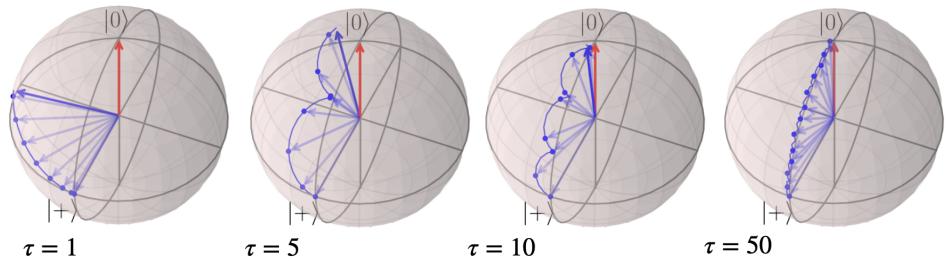


Figure 2.1: Bloch sphere illustration of the single-spin system driven by the Hamiltonian of Eq. (2.1) for different total driving times τ . The red arrow indicates the ground state of the Hamiltonian at $t = \tau$ while the blue path is that taken by the spin during the evolution.

of the field. This moves the spin toward the z axis but also produces a component out of the $x - z$ plane. As the total time for the rotation becomes longer (i.e. the rotation 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, and is 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

Chapter 2. Quantum Adiabaticity

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 eigenstates. 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:

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

where $c(\lambda)$ are time-dependent coefficients through the parameter λ , $\dot{\lambda} = \frac{d\lambda}{dt}$ 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 obtain:

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

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 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 compact and 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:

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

Inserting 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, which is a condition that will be discussed in more detail in the next section, we can neglect it and the solution for the remaining

differential equation of c_m is just:

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

where

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

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. Think of 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 this, 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, 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* [12].

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:

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

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). The integral doesn't can be simplified a lot by using the fact that:

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

where 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:

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

where the last line is a consequence of the assumption that the system starts in the eigenstate m , 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 $e^{-\frac{i}{\hbar\lambda} \int_0^1 \omega_{m,n}(\lambda') d\lambda'}$ and $e^{-i(\gamma_m(\lambda) - \gamma_n(\lambda))}$ (which has been absorbed into $A_{n,m}(\lambda)$) have a maximal value of 1. The same goes for $c_n(1)$. This leaves us with a bound on the remaining quantities:

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

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

Chapter 2. Quantum Adiabaticity

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]. These results are possible because quantum chaos manifests itself through exponential sensitivity of the eigenstates to infinitesimal perturbations that are generated by the AGP, making the scaling of its norm a very sensitive probe for quantum chaos and thermalisation.

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 basis, rewriting the Schrödinger equation reveals something interesting:

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

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’ 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. In fact, the name originates from quantities under which the physics is invariant. 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 $x_0(t)$:

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

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

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

In this thesis we will 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

Chapter 2. Quantum Adiabaticity

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 $\mathcal{A}_\lambda = i\hbar\partial_\lambda$.

2.2.2 Matrix elements of the AGP

What we have seen 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

Chapter 2. Quantum Adiabaticity

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:

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

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 several important questions, starting with *why do we care?* What is it about adiabatic dynamics that makes them important? Why do we want to quantify non-adiabatic transformations and understand what generates? 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 comes 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

Chapter 2. Quantum Adiabaticity

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 $|\psi(\lambda)\rangle = e^{i\dot{\lambda}^{-1}\theta_m(\lambda)} e^{i\gamma_m(\lambda)} |m(\lambda)\rangle$ 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 $H_{t-less}(\lambda)$ (transitionless) that drives these states exactly, we can pick a unitary $R(\lambda)$ such that:

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

It turns out this unitary is just:

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

so the transitionless Hamiltonian can be expressed as:

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

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 than the negation of the AGP component in Eq. (2.33). You may 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

Chapter 2. Quantum Adiabaticity

Hamiltonian:

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

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,

Chapter 2. Quantum Adiabaticity

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 we explore the different ways in which the AGP and thus the counterdiabatic drive has been approximated and what are the advantages and drawbacks of each approach.

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

Chapter 2. Quantum Adiabaticity

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 in 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 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:

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

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

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

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

It turns out that this result can be used to quantify how close some arbitrary operator is to the AGP. In order to do this, we first define an *ansatz* Hermitian operator for \mathcal{A}_λ which acts on the same Hilbert space and which we denote \mathbb{A}_λ . Then, we can define an operator G_λ :

$$G_\lambda(\mathbb{A}_\lambda) = \partial_\lambda H + \frac{i}{\hbar} [\mathbb{A}_\lambda, H]. \tag{2.43}$$

We can see that when $\mathbb{A}_\lambda = \mathcal{A}_\lambda$, i.e. when our guess - or approximation - for the AGP is

Chapter 2. Quantum Adiabaticity

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 $G_\lambda(\mathbb{A}_\lambda)$ and $-F_\lambda$ with respect to the ansatz \mathbb{A}_λ .

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 $G_\lambda(\mathbb{A}_\lambda)$ and $-F_\lambda$ is simply to minimise its Hilbert-Schmidt norm. Let us express this norm as an *action* [4] associated with the AGP:

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

which is minimised whenever \mathbb{A}_λ satisfies:

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

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 $\{\mathcal{O}_{\text{LCD}}\}$ which satisfy the constraints of our physical system. We can then define an approximate AGP in the basis of these operators as:

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

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 $\{\mathcal{O}_{\text{LCD}}\}$, we can think of them as a fixed parameter, and the minimisation procedure consists of minimising the resulting action $\mathcal{S}(\mathbb{A}_\lambda)$ 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 $\tilde{A}_\lambda = i\hbar U^\dagger \partial_\lambda U$ 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 A_λ , which is σ^y :

$$A_\lambda = \alpha(\lambda) \sigma^y. \quad (2.48)$$

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 $G_\lambda(A_\lambda)$ and to minimize the corresponding action $S(A_\lambda)$ with respect to the driving coefficient α . Once this process is complete (see App. 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.

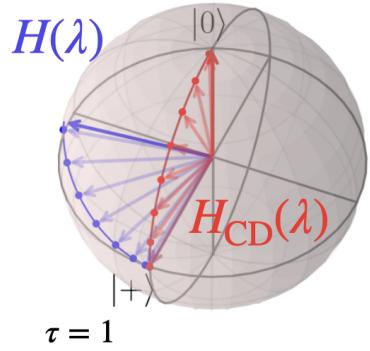


Figure 2.2: State of the rotating spin starting in state $|+\rangle$ driven without CD as in the Hamiltonian of Eq. (A.1) (blue) and with CD as given by Eq. (2.50) (red) for total driving time $\tau = 1$.

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 is quite a sizeable reduction in the true number operators in the full AGP.

2.4.2 Nested commutator expansion

The LCD approach is particularly useful 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. A useful question to ask 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 $H(\lambda)$:

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

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:

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

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 driving 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 with no further insight.

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)

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 [24], this chapter aims to give a broad overview of the topic highlighting

Chapter 3. Quantum Optimal Control

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. As such, in Sec. 3.1, I will begin by exploring the general structure of optimal control problems in detail 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. I will begin with the mathematical structure of optimal control problems (Sec. 3.1.1) followed by an overview and examples of analytic (Sec. 3.1.2) and numerical (Sec. 3.1.3) methods for finding solutions to said problems, with a focus on methods that will be relevant to the rest of the content in this thesis. Sec. 3.2 will review 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. There may be many paths to achieve the target and there may be many metrics to determine what ‘best’ means. The aim of the first part of this section is thus to broadly cover the mathematical structure of optimal control problems and to try and convey an idea of what an optimal path *is* and how one might quantify its optimality. Later in the chapter, I will delve more into practical questions of controllability and the process of optimisation, i.e. once an optimal control problem is constructed, how does one go about finding the solution to it. I will cover both analytic methods in Sec. 3.1.2 and numerical approaches in Sec. 3.1.3 focusing on a select few optimisation algorithms which will be relevant to further chapters of this thesis.

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

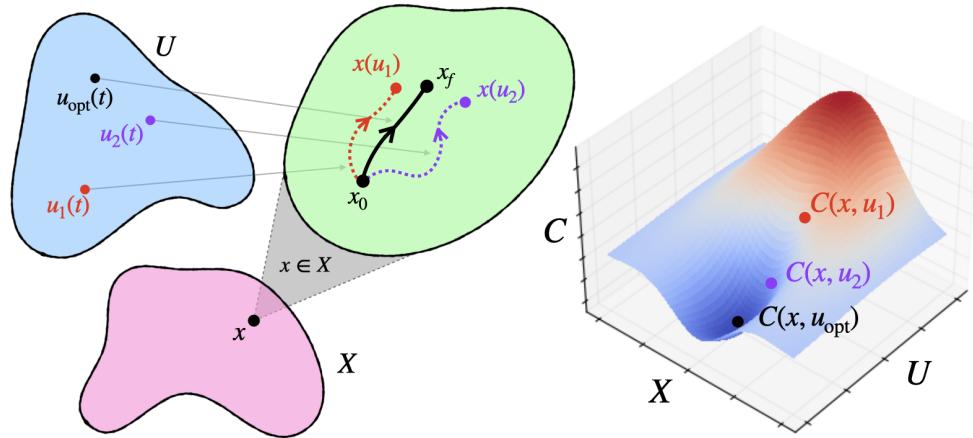


Figure 3.1: Illustration of the Mayer type optimal control problem: when an initial value of the system state x_0 is fixed, the choice of control function $u \in U$ and the requirement of satisfying Eq. (3.1) determine x uniquely. The task is then to find u_{opt} such that the functional $C(x, u_{\text{opt}})$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

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 primarily three different types of problem structures in optimal control centering on different constraints and targets: *Mayer-type*, *Lagrange-type* and their combination, *Bolza* problems [34]. In this thesis, we will mostly focus on Mayer-type problems, particularly in Ch. 4 and Ch. 6. In Mayer problems, the initial state is specified $x(0) = x_0$ and the cost function is of the form

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

with ϕ 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 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 in the case of real vectors over Cartesian

coordinates.

From the above, we can view Mayer-type problems as being concerned primarily with the final state of the system and not its path. Lagrange-type problems, on the other hand, put focus on the behaviour of the system throughout the control trajectory and they encompass cost functions of the type

$$C(u) = \int_0^\tau L(x, u, t) dt, \quad (3.5)$$

where L is a smooth function. This type of cost function is applicable, for example, in cases where one wants to minimise the expenditure of some path-dependent resource during the control procedure, or where a path-dependent quantity is easier to optimise over than a target state quantity. This type of optimisation is something that will become relevant in Ch. 5 and Ch. 7.

The most general type of problem is the Bolza problem, which combines both Mayer and Lagrange in a way that puts emphasis both on the target state of the optimal control and the trajectory that a system takes to get there:

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

where ϕ and L are smooth functions given in Eq. (3.4) a. A great example of Bolza-type problems is the cost function given by Eq. (3.3), which comprises a competition between distance to a target state and the energy expended to drive the system to said 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

Chapter 3. Quantum Optimal Control

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

$$C(u^\epsilon) - C(u) \geq 0, \quad (3.7)$$

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 Appendix B), 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.

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.

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, as a general rule, 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, numerical methods tend to be far more popular than analytic ones simply due to their flexibility and applicability. Where analytic approaches fail, the only way forward is often a numerical method.

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 [43] 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:

$$\mathbf{u}_{n+1} = \mathbf{u}_n - \mu \nabla_{\mathbf{u}} C(\mathbf{u}_n), \quad (3.8)$$

where n denotes the current iteration of the algorithm, $\nabla_{\mathbf{u}} C(\mathbf{u}_n)$ 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 [44], which employ information about the

Chapter 3. Quantum Optimal Control

second derivative to guide the search and provide faster convergence as well as a myriad of other approaches including stochastic methods [45].

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 [46], 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 [47] 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.

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 [48], 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.

Finally, in numerical optimal control we can make 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 (see Fig. 3.4).

The closed loop approach is more resilient to uncertainties and disturbances but requires real-time computation or pre-computed feedback laws. In this thesis, the focus will be exclusively on open-loop approaches, as closed-loop methods require access to live experimental data which was not available in the case of the methods explored in later chapters. However, it is important to acknowledge that the results obtained in open-loop optimisations may not reflect the realistic, complex response a physical system might have to a specific control protocol.

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

3.1.3.1 Nelder-Mead

A frequently used gradient-free optimisers is Nelder-Mead (or downhill-simplex) method [49] 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.

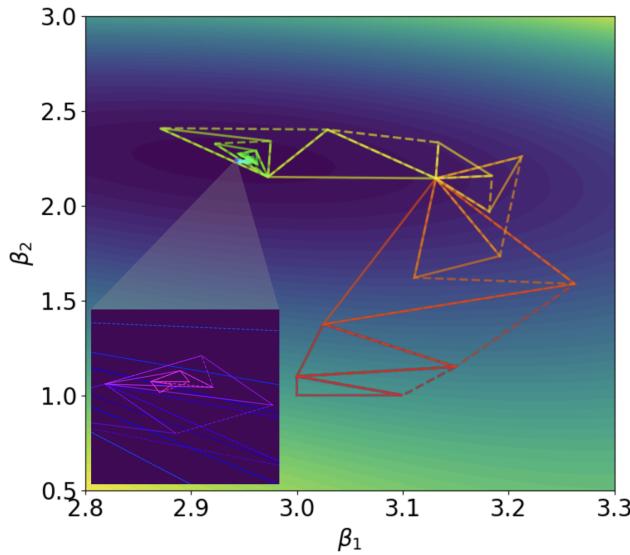


Figure 3.2: Illustration of the Nelder-Mead algorithm for a cost function parameterised by two parameters β_1 and β_2 . The simplex for a 2-dimensional landscape is a triangle, which then follows the rough algorithm described in the text. The iterations of the algorithm are indicated by constantly switching 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.

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 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

Chapter 3. Quantum Optimal Control

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 [50]. The algorithm is known as a *conjugate-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 [51] that the function

$$f(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} - 2\mathbf{b}^T \mathbf{x} + c \quad (3.9)$$

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 $\{u_j\}_{j=1,\dots,n}$ with

$$\beta_i = \frac{\mathbf{u}_i^T \mathbf{b}}{\mathbf{u}_i^T A \mathbf{u}_i}. \quad (3.10)$$

This minimum can be calculated efficiently just through evaluating the cost 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

```

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 [51]. It has guarantees of being very efficient in convex optimisation problems and excels in high-dimensional spaces, unlike Nelder-

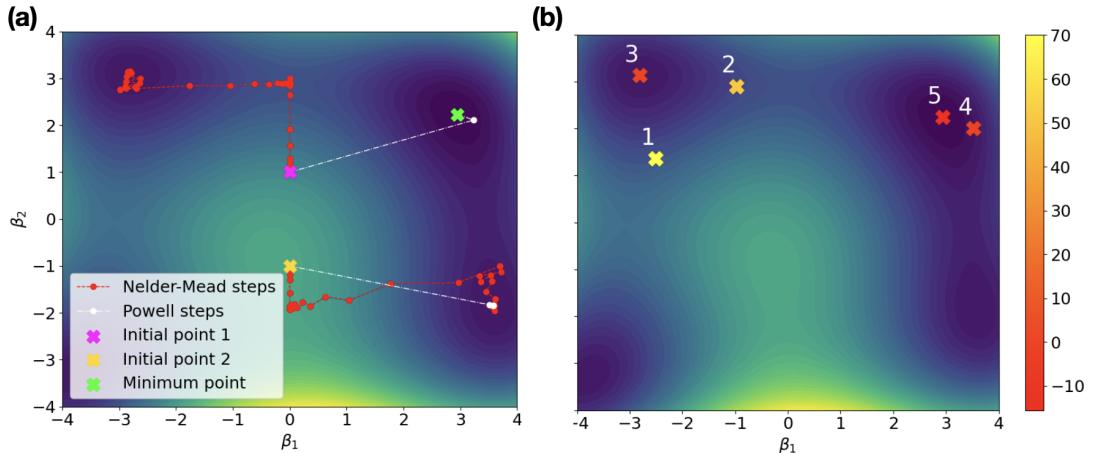


Figure 3.3: An illustration of the optimisation strategy of several numerical optimisation methods. (a) Local optimisers: steps of the Nelder-Mead method and Powell's method when instantiated in two different locations of the loss function landscape. The global minimum is illustrated by a green cross. (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.

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

Chapter 3. Quantum Optimal Control

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) [48], 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 [52] 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_{q_v} . 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 [53]. 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 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 a quantum system. Very broadly then, QOCT concerns itself with the design and analysis of electromag-

Chapter 3. Quantum Optimal Control

netic 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 chapter, I will broadly cover the basics of QOCT, starting with how the mathematical structure discussed in Sec. 3.1.1 can be adapted to the quantum setting and ending with detailed descriptions of CRAB (Sec. 3.3.1) and GRAPE (Sec. 3.3.2), popular QOCT methods which will be relevant to later work presented in this thesis. As the content of later chapters will focus on closed systems, that will be the perspective I will take with respect to QOCT. More concretely, we will stick to cases where the generator of transformations of a quantum system is primarily modelled as the Hamiltonian as opposed to, e.g. a Liouvillian, but a similar analysis holds in the case of open systems which are just as interesting, if not quite as relevant to this thesis.

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:

$$H(\mathbf{u}(t)) = H_0 + \sum_{k=1}^{N_k} u_k(t) \mathcal{O}_{\text{opt}}^{(k)}, \quad (3.11)$$

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 $\mathcal{O}_{\text{opt}}^{(k)}$.

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:

$$i\hbar\partial_t |\psi(t)\rangle = H(\mathbf{u}(t)) |\psi(t)\rangle \quad (3.12)$$

with the constraint of starting in a state from a set of initial states $|\psi_0\rangle \in \Psi_0$ and while

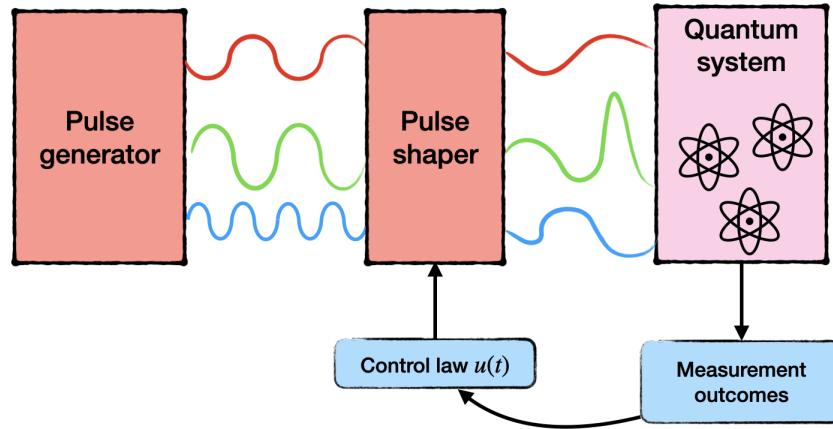


Figure 3.4: A sketch of a quantum optimal control closed-loop set-up. A quantum system is directly controlled by a set of electromagnetic pulses which are shaped according to a set of control functions $\mathbf{u}(t)$ that are optimised based on feedback from the information obtained through measurements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

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

$$\partial_t \mathcal{O}(t) = -iH(\mathbf{u}(t))\mathcal{O}(t), \quad (3.13)$$

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 (3.13) 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 infor-

mation 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:

$$C_F(\tau, \mathbf{u}) = 1 - F(\tau, \mathbf{u}) = 1 - |\langle\psi(\tau, \mathbf{u})|\psi_T\rangle|^2, \quad (3.14)$$

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:

$$C_E(\tau, \mathbf{u}) = \langle\psi(\tau, \mathbf{u})|H_T|\psi(\tau, \mathbf{u})\rangle. \quad (3.15)$$

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:

$$C_S(\tau, \mathbf{u}) = -S[|\psi(\tau, \mathbf{u})\rangle], \quad (3.16)$$

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:

$$C(\tau, \mathbf{u}) = C_F(\tau, \mathbf{u}) + \sum_{k=1}^{N_k} \int_0^\tau |u_k(t)|^2 dt, \quad (3.17)$$

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 [54] or reachability of states, e.g. exploring quantum speed limits [55–57]. 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 [58, 59] 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) [60–62]. It was believed that such systems were mostly intractable for control optimization using gradient-based algorithms [63], although such potential limitations have been overcome in more recent work [64]. 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 :

$$\mathbf{u}(t) = \sum_{i=1}^{N_k} c_i \mathbf{u}_i(t), \quad (3.18)$$

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:

$$\mathbf{u}(g(t), t) = \sum_{i=1}^{N_k/2} c_i \frac{\cos \omega_i t}{g(t)} + \sum_{i=N_k/2+1}^{N_k} c_i \frac{\sin \omega_i t}{g(t)}. \quad (3.19)$$

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

$$\mathbf{u}(t) = f(t) \left(1 + \sum_{i=1}^{N_k} c_i \mathbf{u}_i(t) \right). \quad (3.20)$$

Another particularly useful alteration to the basic CRAB procedure is what is known

Chapter 3. Quantum Optimal Control

as ‘dressed’ CRAB or dCRAB [65], 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

$$u^j(t) = c_0^j u^{j-1}(t) + \sum_{i=1}^{N_k} c_i^j u_i^j(t) \quad (3.21)$$

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 [66] and has since been iterated upon and improved a number of

Chapter 3. Quantum Optimal Control

times as well as being integrated into several optimal control packages [67–70]. As the name suggests, it is a gradient-based optimisation method and while initially it was used primarily for the preparation of specific target states, its powerful flexibility has since lent itself to many other applications in the setting of quantum technologies, like the optimisation of quantum logic gates [71, 72].

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 $\tau = N_m \Delta t$.

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. (3.8)). Recalling the form of the quantum control Hamiltonian from Eq. (3.11), the propagator for the time-evolution of the quantum system using GRAPE during a single time step Δt at time t_k is

$$U_k(\Delta t) = \exp \left\{ -i\Delta t \left(H_0 + \sum_{j=1}^{N_j} u_j(t_k) \mathcal{O}_{\text{opt}}^{(j)} \right) \right\} \quad (3.22)$$

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):

$$U(\tau) = U_{N_m} U_{N_m-1} \dots U_2 U_1, \quad (3.23)$$

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

$$\begin{aligned}\rho(\tau) &= U(\tau)\rho_0U^\dagger(\tau) \\ &= U_{N_m} \dots U_1 \rho_0 U_1^\dagger \dots U_{N_m}^\dagger\end{aligned}\tag{3.24}$$

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. 3.14:

$$C(\mathbf{u}) = \text{Tr}\left\{\rho_T^\dagger \rho(\tau)\right\},\tag{3.25}$$

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. 3.24 and the cyclic property of the trace we can write

$$\begin{aligned}C(\mathbf{u}) &= \text{Tr}\left\{\rho_T^\dagger U_{N_m} \dots U_1 \rho_0 U_1^\dagger \dots U_{N_m}^\dagger\right\} \\ &= \text{Tr}\left\{U_{k+1}^\dagger \dots U_{N_m}^\dagger \rho_T U_{N_m} \dots U_{k+1} U_k \dots U_1 \rho_0 U_1^\dagger \dots U_j^\dagger\right\} \\ &= \text{Tr}\{\Lambda_k \rho_k\},\end{aligned}\tag{3.26}$$

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

$$\delta U_k = -i\delta u_j(t_k) U_k \int_0^{\Delta t} U_k(t') \mathcal{O}_{\text{opt}}^{(j)} U_k(-t') dt'.\tag{3.27}$$

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:

$$\frac{\delta C(\mathbf{u})}{\delta u_j(t_k)} = - \text{Tr} \left\{ \Lambda_k \left(i\Delta t \left[\mathcal{O}_{\text{opt}}^{(j)}, \rho_k \right] \right) \right\}. \quad (3.28)$$

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. 3.8. The method has been improved upon after the initial algorithm was first published, e.g. in [69] 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 [68]. 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 [73]. 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. Furthermore, Krotov-based approaches are only well-defined in the continuum limit i.e. where the time step $\Delta t \ll 1$, which need not be the case for many implementations of GRAPE. In fact, Krotov approaches have been shown to be monotonically convergent in that limit [74], so they can suffer in the face of discretisation, something that is a roadblock for some implementations. GRAPE escapes this fate and, despite in many ways being less sophisticated, can be far more efficient in more restrictive settings with large time steps far away from the continuum limit.

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. There has

been a lot of work done in recent years analyzing the topological and mathematical properties of quantum control landscapes, including their smoothness [75–77], although these only apply to problems where the cost function is “well behaved” - i.e. is some polynomial of the final state vector or matrix which itself evolves continuously on a smooth manifold. 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

I feel a need... a need for speed.

LT Pete "Maverick" Mitchell,
Top Gun, 1986

In Ch. 2 we established that adiabatic evolution of a quantum system requires timescales that scale with the energy gap, without which it experiences non-adiabatic excitations out of its instantaneous eigenstate(s). This presents a problem, as the results of adiabatic dynamics - i.e. keeping the system in its instantaneous eigenstate throughout evolution - are useful in many applications of quantum technologies [3, 21, 78], but the timescales they require are often difficult to achieve due to decoherence and other physical constraints.

The dual motivation of implementing adiabatic evolution and doing so *fast* has led to the development of a number of methods and approaches under the umbrella of shortcuts to adiabaticity (STA) [8], with a universal STA approach being provided by counterdiabatic driving (CD) [6, 7], introduced in detail in Sec. 2.3. However, as established in Sec. 2.3, exact CD is often difficult to derive and even more difficult to implement in an experimental setting, leading to the development of approximate methods such as local counterdiabatic driving [16] (LCD) and the truncated nested-

Chapter 4. Counterdiabatic optimised local driving

commutator approach [17] which were discussed in detail in Sec. 2.4.1 and Sec. 2.4.2 respectively. Apart from the already mentioned techniques, many other approaches [79–81] have been developed which aim to bypass the inherent complexity of the exact CD. These approximate methods all have their advantages and drawbacks when applied to particular adiabatic processes, owing both to their approximate nature and the practical aspects of their implementation.

In this chapter I will present a new method for speeding up adiabatic processes: Counterdiabatic Optimised Local Driving (COLD), which was first developed with the goal of improving upon the results of LCD while retaining the advantages that it offers. Namely: COLD is a method that, given a time-dependent Hamiltonian and a set of physical constraints for the system that is being driven, can be used to construct an approximate counterdiabatic protocol that performs optimally for the given set of constraints on the Hamiltonian and the system. It does this by combining LCD and optimal control, which we explored in detail in Ch. 3. While LCD can be used to implement an approximate CD protocol built out of restricted, physically realisable operators, COLD does this via finding an optimal path for the system, such that the approximate counterdiabatic drive is maximally effective in suppressing non-adiabatic effects.

I will begin the chapter by introducing the COLD method in detail. Then, in Sec. 4.2, I will explore exactly what part QOCT plays in the new method. This chapter lays the groundwork for the method of COLD, while in Part III of the thesis I will present and analyse the results of its numerical implementation in various physical systems.

4.1 Counterdiabatic driving and optimal control

Let us begin by explicitly setting the stage for the problem that we want to solve. Given a Hamiltonian $H(\lambda)$, which depends on time via the parameter $\lambda(t)$, and a system prepared in an eigenstate of $H(\lambda_0)$, where $\lambda_0 = \lambda(0)$ (often this is the ground state, but it need not be), our task is to vary the parameter λ from its initial value

Chapter 4. Counterdiabatic optimised local driving

λ_0 to some final value $\lambda_f = \lambda(\tau)$ during a duration of time τ such that at the end of the process, the system is in the corresponding eigenstate of $H(\lambda_f)$. More precisely, if e.g. the system starts in the ground state of $H(\lambda_0)$, after the evolution it should be in the ground state of $H(\lambda_f)$. This can be done quite reliably, as per the discussion of Ch.2, as long as the instantaneous eigenstates of the Hamiltonian driving the system are not degenerate throughout the evolution and the driving happens slowly enough (see Sec. 2.1.2). However, bearing in mind that such slow evolution is generally not accessible, our primary goal is to achieve this result while keeping τ small i.e. making the evolution as fast as possible.

As already mentioned in the introduction to this Chapter, one way to achieve this task is by using CD (Sec. 2.3). That is, for a given $H(\lambda)$, it may be possible to derive and implement an exact counterdiabatic Hamiltonian from Eq. 2.40 which suppresses all non-adiabatic effects experienced by the system due to fast driving. Exact CD could, in this way, keep a system in the instantaneous eigenstate of $H(\lambda)$ during arbitrarily short driving times (within the geometric speed limit [82]), but exact CD is a commodity that is not generally accessible for an arbitrary Hamiltonian [4], much less implementable in practice, as it often requires highly non-local operators.

The next best thing to try, then, might be an approximate CD method like LCD. As discussed in Sec. 2.4.1, LCD not only allows one to variationally approximate the full CD, thus suppressing some of the losses associated with non-adiabatic effects, but it also gives one the freedom to choose the basis of operators for the approximation, making it very attractive in experimental settings where only a limited set of physical operators are available. If an ansatz is not forthcoming, it is also possible to use the ideas presented in Sec. 2.4.2 to build up a local operator basis which contributes to the full CD via the nested commutator approach [83] and to then use the variational method of LCD in order to construct a counterdiabatic schedule made up of a physically implementable subset of that basis.

The LCD method is powerful, but it is not without its faults. The primary disadvantage of such an approach is that the counterdiabatic drive being implemented will always be an *approximation* unless the ansatz basis is fully representative of the exact

Chapter 4. Counterdiabatic optimised local driving

CD. In cases where the approximation is a poor one, the LCD technique might not offer any suppression of errors at all. One solution to this would simply be to expand the ansatz basis in order to access more degrees of freedom in describing the CD, but this would be counter to the idea of only requiring a physically implementable set of operators as part of the approximation in order to make it useful in an experimental setting. The other would be to use a different method entirely to achieve the same result by, for example, taking a page out of optimal control theory as covered extensively in Ch. 3. It is not obvious, however, that a switch in tactics would lead to an improvement or what the complexity of designing a new approach might be.

This is where we come to the new method, COLD, which was developed with the aim of retaining the advantages of LCD while improving upon its results. The approach begins with the observation that any counterdiabatic schedule will depend on the driving path of the original Hamiltonian for which it is constructed, as discussed extensively in Ch. 2. Namely, if we write a Hamiltonian $H(\lambda)$ as a sum of N_H operators $\{\mathcal{O}_H^{(i)}\}_{i=1,\dots,N_H}$ each scaled by a λ -dependent coefficient $h_i(\lambda) \in \mathbf{h}$ (note that I include constant functions here too instead of treating them as time-independent):

$$H(\lambda, \mathbf{h}) = \sum_{i=1}^{N_H} h_i(\lambda) \mathcal{O}_H^{(i)}, \quad (4.1)$$

then the CD drive can be expressed as a sum of operators $\{\mathcal{O}_{CD}^{(j)}\}_{j=1,\dots,N_{CD}}$ which are scaled by functions $\alpha_j(\lambda, \mathbf{h})$ and the rate of change in the parameters $\dot{\lambda} = \frac{d\lambda}{dt}$. That is to say the form of the counterdiabatic drive is a function of the time-dependent parameter λ , the operators $\mathcal{O}_H^{(i)}$ and their λ -dependent coefficients. Looking back at Eq. (2.40), we can now write the counterdiabatic Hamiltonian as:

$$\begin{aligned} H_{CD} &= H(\lambda, \mathbf{h}) + \dot{\lambda} \mathcal{A}_\lambda \\ &= \sum_{i=1}^{N_H} h_i(\lambda) \mathcal{O}_H^{(i)} + \sum_{j=1}^{N_{CD}} \dot{\lambda} \alpha_j(\lambda, \mathbf{h}) \mathcal{O}_{CD}^{(j)}, \end{aligned} \quad (4.2)$$

where $\{\mathcal{O}_{CD}\}$ is an operator basis of the adiabatic gauge potential \mathcal{A}_λ (AGP) which was introduced at length in Sec. 2.2. We can even see how this relationship comes about

Chapter 4. Counterdiabatic optimised local driving

by looking at the matrix elements of the AGP in Eq. (2.32), which are a function of the instantaneous eigenenergies of $H(\lambda, \mathbf{h})$ and the matrix elements of $\partial_\lambda H(\lambda, \mathbf{h})$, all of which can be written as functions of λ and \mathbf{h} . Note, that in any finite system the operator basis of AGP will be finite.

In this setting, LCD is a way to variationally find the coefficients α_j for a given subset of the full AGP basis $\{\mathcal{O}_{\text{LCD}}\} \subset \{\mathcal{O}_{\text{CD}}\}$ which minimise the operator distance between the generalised adiabatic force from the exact AGP and the force generated by the approximate AGP. In the case where the ansatz is the full basis set $\{\mathcal{O}_{\text{CD}}\}$, one should recover the exact AGP using the variational approach.

The reason for expressing the counterdiabatic Hamiltonian in this way is to make the dependence of the coefficients α_j on the functions \mathbf{h} and λ explicit. As mentioned at the start of this section, the philosophy of COLD begins with the observation that the form of the counterdiabatic drive will depend on the path of the Hamiltonian in the parameter space of its coefficients, i.e. if we change either \mathbf{h} or λ (or both), the form of the counterdiabatic drive will change via the functions α_j . A different way of looking at it is to view each specific set of parameters (\mathbf{h}, λ) as defining a new time-dependent Hamiltonian with its own instantaneous eigenbasis that generates different non-adiabatic effects for a finite evolution time (see, e.g. Eq. (2.24) and the discussion surrounding the AGP).

Let us return to the problem stated at the beginning of this section: our aim is to drive a system which started in an eigenstate of $H(\lambda_0)$ to the corresponding eigenstate of $H(\lambda_f)$ in the shortest amount of time possible. It is important to note that the problem statement does not say anything about the state for any other value of λ throughout the evolution, although in the case where exact CD is implemented, the system should follow the instantaneous eigenstates of H throughout the full dynamics. This more relaxed condition means that, as long as the time-dependent Hamiltonian which drives the system matches up with the problem Hamiltonian at the start and end of the dynamics and the system is in the correct eigenstate at those two points, the path that it takes between them is not particularly important barring any other constraints. This observation can now allow us to finally introduce COLD.

4.1.1 The method

In Sec. 3.2, we delved into the many ways in which driving pulses for quantum systems can be systematically constructed and modified or optimised in order to achieve particular goals. In fact, it is possible to use QOCT to speed up adiabatic protocols too, something that has been studied extensively, with the resulting methods generally grouped under the umbrella of STA [8]Reminder: citations. We could imagine casting out original problem of driving a system to a particular eigenstate of $H(\lambda_f)$ as simply an optimisation problem with a cost function focused on state fidelity given by Eq. (3.14).

In the case of COLD, however, the first step is to construct a control Hamiltonian in the vein of Eq. (3.11) with the constraints that it be equal to $H(\lambda_0)$ at $t = 0$ and $H(\lambda_f)$ at $t = \tau$:

$$H_{\beta}(\lambda, \mathbf{h}, \boldsymbol{\beta}) = \sum_{i=1}^{N_H} h_i(\lambda) \mathcal{O}_H^{(i)} + \sum_{k=1}^{N_k} \beta_k(\lambda) \mathcal{O}_{\text{opt}}^{(k)}. \quad (4.3)$$

Here, $\beta_k(\lambda) \in \boldsymbol{\beta}$ are the control functions which can be constructed and optimised using the methods described in Sec. 3.2 and $\mathcal{O}_{\text{opt}}^{(k)}$ are controllable operators, which can be a subset of $\{\mathcal{O}_H\}$ or introduce a new degree of freedom to the Hamiltonian, as long as the constraints that $H_{\beta}(\lambda_0) = H(\lambda_0)$ and $H_{\beta}(\lambda_f) = H(\lambda_f)$ are satisfied.

In the second step of defining COLD, we return to LCD and the observation that our approximate counterdiabatic drive will depend on the coefficients of the Hamiltonian. If we are given an ansatz set of operators $\{\mathcal{O}_{\text{LCD}}^{(j)}\}_{j=1, \dots, N_{\text{LCD}}}$ and use them to variationally determine the approximate CD protocol for the control Hamiltonian H_{β} of Eq. (4.3), the resulting Hamiltonian will look something like this:

$$\begin{aligned} H_{\text{COLD}}(\lambda, \mathbf{h}, \boldsymbol{\beta}) &= H_{\beta}(\lambda, \mathbf{h}, \boldsymbol{\beta}) + H_{\text{LCD}}(\lambda, \mathbf{h}, \boldsymbol{\beta}) \\ &= \sum_{i=1}^{N_H} h_i(\lambda) \mathcal{O}_H^{(i)} + \sum_{k=1}^{N_k} \beta_k(\lambda) \mathcal{O}_{\text{opt}}^{(k)} + \sum_{j=1}^{N_{\text{LCD}}} \dot{\lambda} \alpha_j(\lambda, \mathbf{h}, \boldsymbol{\beta}) \mathcal{O}_{\text{LCD}}^{(j)}, \end{aligned} \quad (4.4)$$

which is the COLD Hamiltonian.

All that is left now is the third COLD step, which is the optimisation of the coefficients $\beta_k(\lambda)$ using QOCT methods presented in Sec. 3.2. A natural, though not

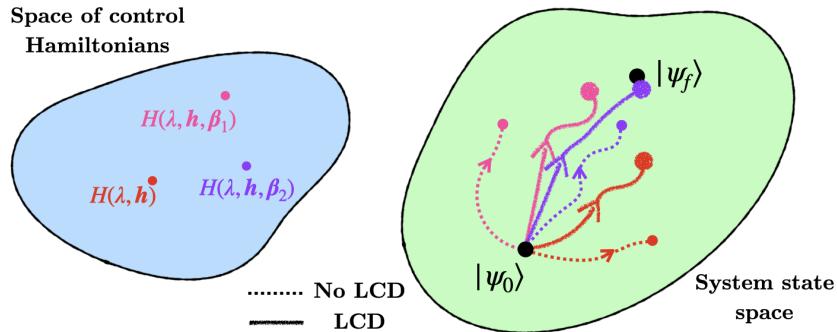


Figure 4.1: A diagrammatic illustration of the COLD method. The blue shape is the set of control Hamiltonians with control parameters β , with each point within the shape representing a different instance of β . In the case where the control amplitude is 0 throughout the evolution, we recover the bare Hamiltonian $H(\lambda, \mathbf{h})$ (red point). The green shape represents the system state space with $|\psi_0\rangle$ the eigenstate of $H(\lambda_0)$ that the system is prepared in and $|\psi_f\rangle$ the corresponding eigenstate of $H(\lambda_f)$, which is the target. The dotted (no LCD drive) and chalky (LCD drive included) directed lines show how the system is driven across state space for a fixed total time τ in the case of each control Hamiltonian indicated on the blue shape.

exclusive, cost function for this process would be the final state fidelity from Eq. (3.14) with respect to the desired eigenstate of $H(\lambda_f)$. I will provide a more detailed discussion of the optimal control component of COLD in the next section.

The addition of the control pulse prior to applying LCD makes it such that the CD coefficients α_j are now functions of β , meaning that varying β will change the shape of the approximate counterdiabatic drive. This is illustrated in Fig. 4.1 with changing paths in state space of the driven system. We take the set of operators $\{\mathcal{O}_{\text{LCD}}^{(j)}\}$ to be fixed, since in a practical scenario this set would depend on physical constraints of the system for which it is implemented. However, the relative contribution of each operator to the exact counterdiabatic pulse (and thus its effectiveness at suppressing non-adiabatic effects) is governed by the path of the Hamiltonian. By adding a control term, we can now optimise the system evolution to follow a path which allows the truncated counterdiabatic drive to maximally suppress non-adiabatic effects. Fig. 4.1 illustrates how the application of LCD will modify any path in system state space to end up closer to the target state than in its absence, but for particular paths it

Chapter 4. Counterdiabatic optimised local driving

will get much closer. Examples of the effectiveness of COLD in different systems are demonstrated and analysed in more detail in Ch. 6, where we explore how COLD holds up against both of its components: LCD and QOCT.

4.2 Optimal control toolbox

With optimal control being one of the two key components of COLD, I will now revisit the content covered in Ch. 3, linking it to the way one might go about constructing control pulses in the COLD setting. In Sec. 3.3 we covered “Chopped Randomised Basis” (CRAB) and “Gradient Ascent Pulse Engineering” (GRAPE), two quantum optimal control methods that offer very flexible yet powerful approaches to constructing and optimising control pulses. Consequently, we can use them in the setting of COLD too. In our original work presented in Ref. [2], three separate techniques for constructing optimal control pulses were implemented which will be the focal point of Part III of the thesis:

- ‘Bare’ pulses, which are functions composed of a Fourier basis where each basis function is scaled by an optimisable coefficient, similar to those given by Eq. 3.19. The name ‘bare’ is used to distinguish them from CRAB as they do not include any randomisation component.
- COLD-CRAB pulses, which are like the bare version but with the inclusion of randomisation in the frequencies of the basis functions used to construct the pulse, as discussed in Sec. 3.3.1.
- COLD-GRAPE pulses, wherein the optimisable function is constructed using the GRAPE approach of parameterised piecewise constant time slices, as was expanded upon in detail in Sec. 3.3.2.

To illustrate, a ‘bare’ pulse, which will make a return often in the next part of the thesis is the function

$$f(\lambda, \beta) = \sum_{k=1}^{N_k} \beta^k \sin(2\pi k \lambda), \quad (4.5)$$

Chapter 4. Counterdiabatic optimised local driving

which fulfills the boundary conditions of $H(\lambda_0)$ and $H(\lambda_f)$ for $\lambda_0 = 0$ and $\lambda_f = 1$. The parameters β_k for each frequency k can be optimised using a numerical approach from Sec. 3.1.3. This is a very simple way to construct a control pulse, but the simplicity is its appeal: it describes a continuous function with very few parameters and a solution can often be easily analysed once it is found.

The case of COLD-CRAB is more self-explanatory, as it is just an implementation of the CRAB algorithm, which was discussed in detail in Sec. 3.3.1, to construct the control pulse for COLD. In the numerical results presented in the next part of the thesis, the most common implementation is taking the bare pulse that defined above in Eq. (4.5) and randomising the principal frequencies $\omega_k = 2\pi k$ of the trigonometric functions. This is done by drawing N_k parameters r_k from a uniform random distribution $r_k \in [-0.5, 0.5]$ at each optimisation instance of the pulse and replacing $k \rightarrow k(1 + r)$. This makes for a far more complicated optimisation process - as already discussed in Sec. 3.3.1 - but it also often allows for far better results without any increase in optimisable parameters. Depending on the computational resources at hand, especially parallelisation, COLD-CRAB is a far better option than just the bare pulse in terms of results.

Finally, COLD-GRAPE is exactly what it says on the tin - if the tin were Sec. 3.3.2. In this case the optimisable pulse is built up out of piecewise constant control amplitudes, as in the original GRAPE algorithm and these are optimised once again using numerical methods from Sec. 3.1.3. The method requires far more optimisation parameters and thus is computationally intensive, but it also removed the need to choose a good pulse basis, like in the bare and COLD-CRAB cases. It is possible, but not necessary, to use the gradient information of the cost function that was provided in the original GRAPE paper [66]. As will become clearer in Ch. 7 however, some of the cost function landscapes we have to deal with in the case of COLD are highly non-convex and as such gradient-based optimisation techniques simply don't work well.

One computational issue to address in using GRAPE for COLD is that the coefficients α_j of the LCD pulses generally have a dependence on $\partial_\lambda \beta$ due to the AGP operator being a function of the matrix elements of $\partial_\lambda H$. These are not well-defined

for a pulse constructed out of piecewise constant amplitudes masquerading as a continuous function. The way to get around this issue is to use spline interpolation [84], which is a method used to interpolate between the piecewise components and recover a continuous pulse which can then be used to calculate the derivatives $\partial_\lambda \beta$.

These three methods are by no means the only way to construct optimal pulses for COLD and, as discussed in Sec. 3.2, the field of quantum optimal control is vast. A consideration that is specific to COLD is the inclusion of constraints in the cost function on the LCD pulse as well as the control drive, as it can diverge e.g. across phase transitions [85], something that makes experimental implementation difficult and goes against the philosophy of COLD as a method. At the end of the day, both the LCD pulse and the control drive should be designed with the goal of making them useful in an experimental setting and the optimal control component needs to reflect this.

Chapter 5

Adiabatic gauge potential as a cost function

Always remember, however, that there's usually a simpler and better way to do something than the first way that pops into your head.

Donald Knuth

In the previous chapter I presented the counterdiabatic optimised local driving or COLD method, which combines LCD (Sec. 2.4.1) and quantum optimal control (Sec. 3.2) in order to speed up adiabatic quantum processes while minimising transitions out of the instantaneous eigenstates. The strategy of COLD is largely concerned with implementing optimal control in order to modify the path of the time-dependent Hamiltonian in parameter space in a way that maximises the effectiveness of a given LCD drive in driving a system to a target eigenstate of the adiabatic Hamiltonian. The optimal control component of COLD is thus constructed around optimising for the final state of the system after evolution: whether by assessing its fidelity with respect to some target state or a property like entanglement.

In this chapter we will take a slightly different but complementary perspective on combining LCD and optimal control by asking the question of what happens when,

Chapter 5. Adiabatic gauge potential as a cost function

instead of optimising for a particular target state, we use only information about the counterdiabatic drive - or rather, the adiabatic gauge potential (AGP) from Sec. 2.2 - as the optimal control cost function. Since the AGP contains information about the non-adiabatic effects experienced by a system, it is reasonable to believe that this information can be extracted and its analysis can be useful in designing optimal fast driving schedules for adiabatic protocols.

I will begin the chapter with a brief motivation behind using the counterdiabatic pulse as a metric for optimising fast adiabatic processes in Sec. 5.1 and then in Sec. 5.2 we will explore several different ways in which the CD pulse can be transformed into an optimal control cost function. While this chapter will introduce the theory behind the idea, Ch. 7 will present the numerical simulation results obtained using the ideas in this chapter.

5.1 Motivation

Returning to the key ideas behind CD, we may recall that the exact CD pulse is comprised of the AGP (Sec. 2.2) operator \mathcal{A}_λ scaled by the rate of change of $\lambda(t)$ (expressed as $\dot{\lambda}$) in the adiabatic Hamiltonian, as given by Eq. (2.40). The AGP is the generator of adiabatic deformations between quantum eigenstates, and its off-diagonal elements are responsible for transitions between the instantaneous (or adiabatic) eigenstates or, put another way, the Frobenius norm of the AGP is the distance between nearby eigenstates [18, 19] @ Callum this is correct, I've changed the references to where this exact thing is stated exactly like this (and I have more) and described in detail. Is it the wording that's confusing somehow?. Thus, there are two components comprising the non-adiabatic effects experienced by a system driven at finite time by a time-dependent Hamiltonian: the rate of change of the parameters $\dot{\lambda}$ and the AGP operator, which can be quantified e.g. via its norm.

In the design fast adiabatic protocols such as the techniques under the umbrella of STA, we are competing against $\dot{\lambda}$ as the aim in general is speed rather than adherence to the adiabatic condition. This leaves us with minimising \mathcal{A}_λ or else, as in the case of CD, suppressing or mitigating its effects. To whit, LCD does this by implementing

Chapter 5. Adiabatic gauge potential as a cost function

an operator that approximately suppresses the AGP for a given Hamiltonian. COLD then aids in this endeavour by modifying the Hamiltonian and thus the corresponding AGP operator in a way that allows for a given LCD protocol to perform better. The optimal control component of COLD in [1] and throughout Ch. 6 is implemented with the target state fidelity (Eq. (3.14)) as a cost function, which is down to the fact that the primary application of adiabatic protocols is often (though not always [23]) state preparation. The use of fidelity as a cost function, however, necessitates access to the wavefunction of the final state, something that becomes difficult to compute for large or highly correlated systems and may result in a highly non-convex or complex cost function landscape (see, for example, Fig. 7.5). Furthermore, fidelity is not a useful cost function in practice in the case where the target state is unknown, which is common in e.g. applications of adiabatic protocols to Hamiltonians whose ground states encode solutions to combinatorics problems [21, 86].

A solution to the problem of fidelity as a cost function might be the use of a different metric for optimising the Hamiltonian path. We have established that \mathcal{A}_λ contains information pertaining to non-adiabatic losses experienced by a driven system, thus a natural approach to optimising the path of the Hamiltonian would be to minimise the AGP operator (I will discuss the details of this in the next section), as this should in principle minimise losses associated with non-adiabaticity. The main advantages of such an approach would be the fact that the minimisation should be far more efficient than any attempt to compute the full system evolution, assuming we have access to the AGP as a function of the Hamiltonian path in parameter space. Furthermore, this process would require no knowledge of the system wavefunction at any point, removing the drawbacks discussed earlier concerning the fidelity cost function.

5.2 Designing a cost function around the counterdiabatic pulse

There are several ways to define a metric on a time-dependent pulse and in this thesis we will explore two in particular. In order to do this, we will first return to Ch. 4, where

Chapter 5. Adiabatic gauge potential as a cost function

we expressed the CD pulse as a sum of operators $\{\mathcal{O}_{\text{CD}}^{(j)}\}_{j=1,\dots,N_{\text{CD}}}$ which are scaled by coefficients $\alpha_j(\lambda, \mathbf{h})$ with λ a function of time and \mathbf{h} the λ -dependent coefficients of the Hamiltonian $H(\lambda)$. In the LCD case, this sum is truncated to a set of operators $\{\mathcal{O}_{\text{LCD}}\} \subset \{\mathcal{O}_{\text{CD}}\}$ which are themselves scaled by a different set of coefficients $\alpha'_j(\lambda, \mathbf{h})$. Expressed this way, we can view the operators as static, with the λ - and \mathbf{h} -dependent coefficients α_j and α'_j encoding the shape of the pulse and containing the information about non-adiabatic effects for given values of λ and \mathbf{h} . If we include an optimal control component parameterised by functions $\beta_k(\lambda) \in \boldsymbol{\beta}$ as in Eq. (4.3) in the case of COLD, then the counterdiabatic coefficients become dependent on $\boldsymbol{\beta}$ too, allowing us to optimise them by varying the control functions.

Given this form for the counterdiabatic pulses, we can choose two types of metrics: one which looks at the whole pulse, whether in the case of the exact AGP or some truncation obtained using LCD and another which instead only picks out an extremum of the pulse, like its maximum amplitude. In the first case, a natural option would be the time-integrated absolute value of the α_j coefficients, which in the optimal control community is often referred to as the “control effort” [87]:

$$C_I(\tau, \boldsymbol{\beta}) = \sum_j^{N_I} \int_0^\tau dt' |\alpha'_j(\lambda(t'), \mathbf{h}, \boldsymbol{\beta})|, \quad (5.1)$$

where, as before, τ is the total driving time and here we take the sum over N_I coefficients. When the quantity $C_I(\tau, \boldsymbol{\beta})$ is minimised, naturally the contribution of the operators scaled by the coefficients in the sum will be reduced. When it is 0, then these operators will not contribute at all to the non-adiabatic losses experienced by the driven system. It should be noted that understanding the physical meaning behind non-zero values of C_I is quite non-trivial, although it is natural to expect that the larger it is, the more losses the system experiences.

In the case where we want to minimise the exact AGP, $N_I = N_{CD}$. On the other hand, when it comes to LCD it may be fruitful to minimise α'_j corresponding to operators which are not actually applied to the system. What this means is that should one have access to a set of operators that have a non-zero contribution to the exact

Chapter 5. Adiabatic gauge potential as a cost function

CD, e.g. via the nested commutator approach from Sec. 2.4.2, but it is only possible to physically implement a subset of these, then it is sensible to optimise for a path which minimises the contribution of the subset of operators that cannot be implemented, as this should theoretically reduce their contribution to the non-adiabatic losses. In this way, an LCD protocol can be used to both suppress non-adiabatic losses by the application of an approximate counterdiabatic drive and in order to optimise for a Hamiltonian path which reduces the non-suppressed losses experienced by the system.

The second option for a cost function which uses nothing but the counterdiabatic pulse coefficients is one which instead minimises some extremum of the entire pulse, such as the maximal absolute amplitude reached by the pulse throughout the evolution, which we will write as:

$$C_A(\tau, \beta) = \max_{t' \in [0, \tau]} \left(\sum_j^{N_I} |\alpha_j(\lambda(t'), \mathbf{h}, \beta)| \right). \quad (5.2)$$

Intuitively, if the integral cost function is 0, then this second approach does not provide any more information as the two will be equivalent. However, what this cost function captures is whether or not the path of the system for a given β and τ experiences any critical point in its evolution where the non-adiabatic effects are maximised, for example in the case of closing gaps between the instantaneous eigenstates. This cost function may be useful in cases where one wants to avoid such closing gaps within the system evolution and no path optimally suppresses all non-adiabatic effects. In particular, it might be used in LCD as a means to avoid cases where the CD operators which are not applied are responsible for most of the avoided transitions out of the instantaneous eigenstate.

Part III

Applications of COLD

Chapter 6

Optimising for properties of the state

In theory, theory and practice
are the same. In practice, they
are not.

Unknown

In Ch. 4 we introduced COLD, a new method for speeding up adiabatic processes while suppressing non-adiabatic losses. In this chapter, we will investigate how such a method might perform for different adiabatic protocols in different physical systems via results from numerical simulations.

There are a number of parameters that can be varied in each instance of applying COLD, including different ways to construct the control pulse (Sec. 4.2), physical constraints placed on the system and the operator basis used for the LCD, among others. Furthermore, it is important to compare the effects of COLD against either of its two components: LCD and quantum optimal control, which have been implemented with the same goal as COLD in the past [8, 16, 31]. We made the claim, in Ch. 4, that COLD should outperform either approach simply by construction and here we will demonstrate this in practice.

We will begin with an example of a two-spin annealing process in order to illustrate the COLD approach in detail, as well as the variational method for deriving an LCD

Chapter 6. Optimising for properties of the state

pulse and constructing an optimal control pulse. We will then proceed to illustrate how COLD can be applied for a series of different example Hamiltonians and systems, starting with the Ising spin chain in Sec. 6.2, then a case of population transfer in a synthetic lattice via an adiabatic rapid passage (ARP) protocol in Sec. 6.3 and finally in preparing maximally entangled states in a system of frustrated spins in Sec. 6.4. We will demonstrate how driving amplitude constraints affect the performance of COLD and other approaches in Sec. 6.2.1 and how different optimisation cost functions can inform the results in Sec. 6.4.1 where we will use entanglement as an optimisation metric instead of final state fidelity.

6.1 Two-spin annealing

To showcase and explore the use of COLD in a relatively simple setting we will consider a two spin quantum annealing problem with Hamiltonian

$$H_0(\mathbf{h}, \lambda) = J(\lambda)\sigma_1^z\sigma_2^z + Z(\lambda)(\sigma_1^z + \sigma_2^z) + X(\lambda)(\sigma_1^x + \sigma_2^x), \quad (6.1)$$

where the operator subscripts denote the index of the spin on which they act, H_0 is parameterised by the functions $\mathbf{h} = \{J(\lambda), Z(\lambda), X(\lambda)\}$, with the $\lambda(t) = \frac{t}{\tau}$ term encoding the time-dependence and where $J(\lambda) = -2J_0$ and $Z(\lambda) = -h_0$ are constant functions. For this example we use

$$X(\lambda) = 2h_0 \sin^2\left(\frac{\pi}{2} \sin^2\left(\frac{\pi}{2}\lambda\right)\right), \quad (6.2)$$

where we note that since $\lambda(0) = 0$ and $\lambda(\tau) = 1$, the transverse field is tuned from 0 to $2h_0$ as t goes from 0 to τ . We consider the case where $J_0/h_0 = 0.5$, meaning that the initial ground state of the system is in the $|\uparrow\uparrow\rangle$ state and the ground state at $t = \tau$ should be a superposition of all the symmetric states.

As per the discussion in Sec. 2.4.1, since H_0 is a real-valued Hamiltonian, the single-spin LCD operators should be fully imaginary and thus given by the following

ansatz for the adiabatic gauge potential:

$$\mathbb{A}_\lambda^{(1)}(\lambda, \mathbf{h}) = \alpha(\lambda, \mathbf{h})(\sigma_1^y + \sigma_2^y), \quad (6.3)$$

which we will indicate as ‘first-order’ or FO LCD, referring to the fact that these are the most local spin operators and denoting this fact with the superscript (1). In this case, we expect the LCD pulse to be the same for both σ_1^y and σ_2^y due to the symmetry of the Hamiltonian, hence only requiring one coefficient α for both operators. In future sections where this is not the case, we will instead differentiate between global and local LCD pulses, where unique pulses may be required for each separate operator, but due to physical constraints we approximate them with a single coefficient.

Using the methods described in [16] and summarised in Sec. 2.4.1, we can determine the form of the coefficient α using a variational approach, which I will set out in detail here to illustrate the method. For the given H_0 , and \mathbb{A}_λ , the first step is to find the operator G_λ from Eq. 2.43:

$$\begin{aligned} G_\lambda(\mathbb{A}_\lambda^{(1)}, H_0) &= \partial_\lambda H_0 + \frac{i}{\hbar} [\mathbb{A}_\lambda^{(1)}, H_0] \\ &= \partial_\lambda X(\sigma_1^x + \sigma_2^x) - 2\alpha J(\sigma_1^x \sigma_2^z + \sigma_1^z \sigma_2^x) - 2\alpha Z(\sigma_1^x + \sigma_2^x) \\ &\quad + 2\alpha X(\sigma_1^z + \sigma_2^z), \end{aligned} \quad (6.4)$$

where the λ -dependence is omitted and the only non-constant function is $X(\lambda)$, making it the only non-zero contribution to $\partial_\lambda H_0$. We then obtain the action from Eq. (2.44) defined as $\mathcal{S} = \text{Tr}[G_\lambda^2]$:

$$\frac{1}{8} \text{Tr}[G_\lambda^2] = 4\alpha^2 X^2 + (\partial_\lambda X - 2\alpha Z)^2 + 4\alpha^2 J^2, \quad (6.5)$$

which can be minimised with respect to the coefficient α in order to find the LCD pulse:

$$\begin{aligned} \frac{\partial \mathcal{S}}{\partial \alpha} &= 8X^2\alpha - 4Z(\partial_\lambda X - 2\alpha Z) + 8J^2\alpha = 0 \\ \Rightarrow \alpha &= \frac{1}{2} \frac{Z\partial_\lambda X}{X^2 + Z^2 + J^2}, \end{aligned} \quad (6.6)$$

where we find, as expected, that the LCD pulse is a function of λ and the Hamiltonian

Chapter 6. Optimising for properties of the state

coefficients \mathbf{h} (and their derivatives with respect to λ). For completeness, the full LCD Hamiltonian, recalling Eq. (2.40), then reads:

$$H_{\text{LCD}}(\mathbf{h}, \lambda) = H_0(\mathbf{h}, \lambda) + \dot{\lambda}\alpha(\mathbf{h}, \lambda)(\sigma_1^y + \sigma_2^y), \quad (6.7)$$

where the counterdiabatic pulse is simply the (approximate) AGP scaled by the rate of change in the time-dependent parameter λ .

We can do the same as above for the next most local, two-spin operators, which we will refer to as ‘second-order’ or SO LCD. They too should be imaginary and, due to the Hamiltonian symmetry, can be cast into two groups with two different LCD coefficients γ and ζ :

$$\mathbb{A}_\lambda^{(2)}(\mathbf{h}, \lambda) = \gamma(\mathbf{h}, \lambda)(\sigma_1^x\sigma_2^y + \sigma_1^y\sigma_2^x) + \zeta(\mathbf{h}, \lambda)(\sigma_1^z\sigma_2^y + \sigma_1^y\sigma_2^z). \quad (6.8)$$

These can be solved for in a similar vein to the method for α , although in this case the minimisation of $\mathcal{S}(\mathbb{A}_\lambda^{(2)})$ will happen separately for both γ and ζ , giving a coupled set of equations, which can be solved numerically. Since the whole system is only made up of two spins, the two orders of LCD ansatz are enough to characterise the AGP fully as they contain all completely imaginary orthogonal operators in the Pauli basis. Thus, solving the coupled set of equations

$$\begin{pmatrix} 2(X^2 + Z^2 + J^2) & -2JX & 4JZ \\ -XJ & X^2 + 4Z^2 & -3XZ \\ 4JZ & -6ZX & 2J^2 + 2Z^2 + 8X^2 \end{pmatrix} \begin{pmatrix} \alpha \\ \gamma \\ \zeta \end{pmatrix} = \begin{pmatrix} Z\partial_\lambda X \\ 0 \\ J\partial_\lambda X \end{pmatrix} \quad (6.9)$$

for the coefficients α , γ and ζ should give the exact AGP operator

$$\mathcal{A}_\lambda(\mathbf{h}, \lambda) = \alpha(\mathbf{h}, \lambda)(\sigma_1^y + \sigma_2^y) + \gamma(\mathbf{h}, \lambda)(\sigma_1^x\sigma_2^y + \sigma_1^y\sigma_2^x) + \zeta(\mathbf{h}, \lambda)(\sigma_1^z\sigma_2^y + \sigma_1^y\sigma_2^z) \quad (6.10)$$

for the Hamiltonian $H_0(\mathbf{h}, \lambda)$ given any \mathbf{h} and λ .

The different approaches are demonstrated in Fig 6.1(a) via numerical simulations of the system evolution for different total evolution times $\tau(h_0^{-1})$. We compare the

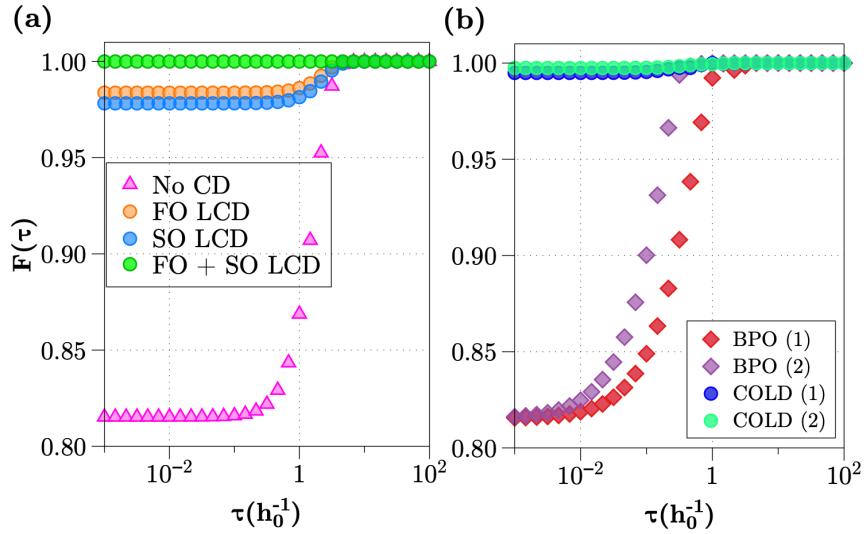


Figure 6.1: Optimisation of the annealing protocol for two spin Hamiltonian given by Eq. (6.1) and with parameters as described in the main text. (a) Final fidelities of the annealing protocol with triangles (pink) representing the case where no CD is applied and circles showing the case of FO LCD (orange), SO LCD (blue) as well as the combination of FO and SO LCD (green). (b) Final fidelities achieved when using the optimal control method BPO ($N_k = 1$: red diamonds, $N_k = 2$: purple diamonds) and COLD ($N_k = 1$: blue circles, $N_k = 2$: aquamarine circles) with FO LCD operators as described in the text.

final evolved state $|\psi(\tau)\rangle$ of the system with the ground state of $H_0(\lambda = 1)$ denoted by $|\psi_{GS}\rangle$ by computing their fidelity:

$$F(\tau) = |\langle\psi_{GS}|\psi(\tau)\rangle|^2. \quad (6.11)$$

The results are computed for the case where only the bare Hamiltonian $H_0(\mathbf{h}, \lambda)$ from Eq. 6.1 drives the system and they are then compared to FO LCD (Eq. (6.7)), SO LCD (the solution to Eq. (6.9) with α set to 0) and the exact counterdiabatic drive (Eq. (6.10)). The results show that at fast evolution times ($\tau < 1 h_0^{-1}$) when no counterdiabatic drive is applied the final state remains far from the ground state of $H_0(\lambda = 1)$ and only begins to approach the target as the evolution time is increased, exactly as one might expect given the adiabatic condition (Sec. 2.1.2). In the case of both FO and SO LCD, however, the system approaches the desired ground state with close to

Chapter 6. Optimising for properties of the state

unit fidelity even at very short driving times several orders of magnitude faster. As expected, when the exact CD is applied including all single- and two-spin imaginary operators, the system reaches the desired state with unit fidelity at arbitrarily short driving times, as in the rotating spin example we discussed in Ch. 2 and Appendix A.

At this point all we have done is to implement the LCD method for a simple example where the exact CD can be easily derived. This was done primarily to illustrate how the LCD pulse is constructed and how it can be used to significantly speed up system evolution while driving it close to the target state by suppressing a large proportion of non-adiabatic losses. With these components explored in depth, we can now finally introduce COLD in a practical setting.

As the goal of COLD is to improve LCD in restricted settings, the most natural approach to demonstrate it in a simple example like this is to take the FO LCD ansatz as a fixed approximation for the counterdiabatic drive and to ignore SO terms. This is a realistic scenario, as even two-spin operators like those found in the SO LCD anstaz of Eq. (6.8) are generally difficult to engineer in physical systems [Reminder: does this need a citation?](#) and thus even in this simple case it is unlikely that the exact CD pulse could be implemented, though its functional form is known.

Revisiting Ch. 4, we find that the first step of COLD is to construct a control pulse for the Hamiltonian. Ideally, this is something that can be easily controlled in an experimental setting and for the given H_0 , we can imagine introducing a λ -dependent control for the coupling, transverse or longitudinal fields. For simplicity and pedagogy, we can introduce a ‘bare’ pulse (Sec. 4.2) control component to the Hamiltonian which drives the $(\sigma_1^z + \sigma_2^z)$ operators and obeys the constraint of being 0 at the beginning and end of the driving time. This gives a control Hamiltonian:

$$\begin{aligned} H_\beta(\boldsymbol{\beta}, \mathbf{h}, \lambda) &= H_0(\mathbf{h}, \lambda) + \sum_{k=1}^{N_k} \beta_k(\lambda)(\sigma_1^z + \sigma_2^z) \\ &= H_0(\mathbf{h}, \lambda) + \sum_{k=1}^{N_k} c_k \sin(\pi k \lambda)(\sigma_1^z + \sigma_2^z), \end{aligned} \tag{6.12}$$

where $\beta_k(\lambda) \in \boldsymbol{\beta}$, $\beta_k(\lambda) = c_k \sin(\pi k \lambda)$, the value N_k denotes the total number of control

Chapter 6. Optimising for properties of the state

functions and the parameters c_k can be optimised using a numerical optimal control method introduced in Sec. 3.1.3. In this case, we will implement Powell optimisation (Sec. 3.1.3.2) as it is an efficient, gradient-free method that heuristically appears to avoid local minima in the cost function space better than Nelder-Mead (Sec. 3.1.3.1), although both can be used given the relatively simple control problem at hand.

Since the control Hamiltonian includes additional non-trivial λ -dependent components that H_0 did not, we need to re-derive the updated FO LCD pulse for the control Hamiltonian H_β , which is not particularly difficult if we follow the earlier recipe:

$$\alpha(\boldsymbol{\beta}, \mathbf{h}, \lambda) = \frac{1}{2} \frac{(Z + f_{\text{opt}}^{N_k})\partial_\lambda X - (\partial_\lambda f_{\text{opt}}^{N_k})X}{(Z + f_{\text{opt}}^{N_k})^2 + X^2 + J^2}, \quad (6.13)$$

where $f_{\text{opt}}^{N_k} = \sum_{k=1}^{N_k} \beta_k$ is the full control pulse constructed out of the N_k components.

The total COLD Hamiltonian with FO LCD is then:

$$H_{\text{COLD}}(\boldsymbol{\beta}, \mathbf{h}, \lambda) = H_0(\mathbf{h}, \lambda) + f_{\text{opt}}^{N_k}(\boldsymbol{\beta}, \lambda)(\sigma_1^z + \sigma_2^z) + \dot{\lambda}\alpha(\boldsymbol{\beta}, \mathbf{h}, \lambda)(\sigma_1^y + \sigma_2^y). \quad (6.14)$$

All that remains is to optimise the parameters c_k with Powell’s optimisation algorithm. As our goal is to drive the system to the ground state of $H_0(\lambda = 1)$, we can implement the fidelity cost function from Eq. (3.14) as a metric for the optimisation. Fig. 6.1(b) shows the results of numerical simulations in the same regime as for the optimisation-free case, comparing evolution under the optimal control Hamiltonian from Eq. (6.12), an approach we call ‘Bare Powell Optimisation’ or BPO and the COLD Hamiltonian (Eq. (6.14)) with a FO LCD pulse included. Even for a single optimisable parameter, COLD achieves $\sim 99.5\%$ fidelity at arbitrarily short times, while BPO remains stuck below $\sim 85\%$ until $\tau \sim 0.1h_0^{-1}$. Both approaches show some improvement with an added control parameter ($N_k = 2$), with the COLD result starting at $\sim 99.8\%$ fidelity even at short times. Neither approach, however, shows any noticeable improvement when adding more control parameters beyond this.

Before moving on to the next section, I want to note that this example is very simple and largely pedagogical. It may be possible to improve the results of both BPO and

Chapter 6. Optimising for properties of the state

COLD with more sophisticated optimal control techniques, but it shows that even in the simplest case, COLD is a powerful method. If reliable access to the operators making up the exact CD is available, then it is better to implement CD rather than attempting optimal control. However, as this is almost never the case given the complexity and non-locality of the exact AGP, COLD is the best way to make the most out of the limited counterdiabatic capacity available.

6.2 Ising chain

A more complex and widely studied example system that we can apply COLD to is the one-dimensional Ising spin chain for N spins in the presence of a transverse and longitudinal field. The Ising model is a fundamental model in quantum mechanics and its ground states can be used to encode solutions to many combinatorics problems [21, 22, 88]. As such, studying annealing protocols for Ising Hamiltonians with arbitrary connectivity is of particular interest. While in this section we will look only at the restricted case of the Ising chain, in Appendix C I derive the coupled equations for FO and SO LCD coefficients for a model with arbitrary $\sigma^z\sigma^z$ connectivity between the spins.

The Ising chain is described by the Hamiltonian

$$H_0(\mathbf{h}, \lambda) = J(\lambda) \sum_j^{N-1} \sigma_j^z \sigma_{j+1}^z + Z_0 \sum_j^N \sigma_j^z + \lambda X_f \sum_j^N \sigma_j^x, \quad (6.15)$$

where once again $\lambda(t) = t/\tau$, $\mathbf{h} = \{J(\lambda), Z(\lambda), X(\lambda)\}$ with constant functions $J(\lambda) = -J_0$, $Z(\lambda) = Z_0$ and

$$X(\lambda) = X_0 \sin^2 \left(\frac{\pi}{2} \sin^2 \left(\frac{\pi}{2} \lambda \right) \right). \quad (6.16)$$

In this section, all results are obtained using $J_0 = 1$, $Z_0 = 0.02J_0$ and $X_0 = 10J_0$.

Many of the steps in this section will be rehashed from the two spin example as the approach is very similar. In this case, we will only focus on the FO LCD terms, which are the single-spin σ^y operators applied to each spin in the chain, in the same vein as

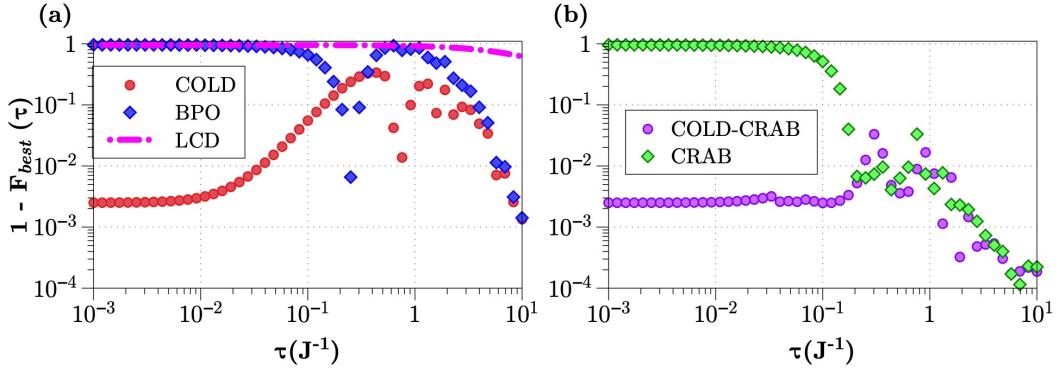


Figure 6.2: Final state fidelities for the Ising spin chain example described in the text for $N = 5$ spins. In (a) we plot the best ($1 - F(\tau)$) obtained from 500 optimisations for different evolution times τ in the case where only FO LCD is applied (pink dash-dot line), as well as BPO (Eq. (6.19), blue diamonds) and COLD with FO LCD (red circles) with a bare pulse control as described in the text. (b) shows implementations of CRAB and COLD-CRAB, with the best result again chosen from 500 optimisations, each with a different randomised frequencies in the trigonometric basis as described in the text. All control pulses use $N_k = 1$ optimisable parameter. Figure reproduced from [1].

in the two spin example of Eq. 6.3 due to the Hamiltonian being real:

$$\mathbb{A}_\lambda^{(1)}(\mathbf{h}, \lambda) = \alpha(\mathbf{h}, \lambda) \sum_j^N \sigma_j^y, \quad (6.17)$$

where using the variational LCD approach we find that

$$\alpha = \frac{1}{2} \frac{Z \partial_\lambda X}{X^2 + Z^2 + 2(1 - 1/N)J^2}, \quad (6.18)$$

with the N -dependent factor in the denominator is a consequence of the edge effects of the chain which disappear in the case of a ring.

In order to implement COLD we need to once again construct a control pulse. In this case, we will start with a similar naive bare pulse as in the two spin case from the

previous section

$$\begin{aligned} H_{\beta}(\boldsymbol{\beta}, \mathbf{h}, \lambda) &= H_0(\mathbf{h}, \lambda) + \sum_k^{N_k} \beta_k(\lambda) \sum_j \sigma_j^z \\ &= H_0(\mathbf{h}, \lambda) + \sum_k^{N_k} c_k \sin(\omega_k \lambda) \sum_j \sigma_j^z, \end{aligned} \quad (6.19)$$

with $\omega_k = 2\pi k$ the k th principal frequency. In the case where no LCD is added to this Hamiltonian, we will again refer to the method as BPO as the numerical optimisation is carried out using Powell's method.

To construct the COLD Hamiltonian, the only additional step is to include the LCD pulse which we will restrict to single-spin terms as in Eq. (6.17):

$$H_{\text{COLD}}(\boldsymbol{\beta}, \mathbf{h}, \lambda) = H_0(\mathbf{h}, \lambda) + f_{\text{opt}}^{N_k}(\boldsymbol{\beta}, \lambda) \sum_j \sigma_j^z + \dot{\lambda} \alpha(\boldsymbol{\beta}, \mathbf{h}, \lambda) \sum_j \sigma_j^y, \quad (6.20)$$

where again $f_{\text{opt}}^{N_k} = \sum_k^{N_k} \beta_k(\lambda)$ represents the full control pulse made up of N_k functions.

As well as the bare pulse, however, in this case we will also implement the CRAB algorithm which was first introduced in Sec. 3.3.1 and compare it to the naive approach. The COLD algorithm with a CRAB-type pulse is thus referred to as COLD-CRAB. In our case, the inclusion of CRAB simply necessitates adding a randomised component to the naive basis functions $\beta_k(\lambda)$. Namely, the pulse $f_{\text{opt}}^{N_k}$ is replaced by $f_{\text{CRAB}}^{N_k}$ where the principal frequencies k are modified as $k \rightarrow k(1+r_k)$, with r_k drawn from a uniform random distribution $r_k \in [-0.5, 0.5]$ such that:

$$f_{\text{CRAB}}^{N_k} = \sum_k^{N_k} c_k \sin(2\pi k(1+r_k) \lambda). \quad (6.21)$$

Since each optimisation instance for the CRAB algorithm will implement a slightly different pulse for the same number of control parameters owing to the randomised component, it is liable to lead to cost functions that have both better and worse minima than those of the naive pulse, meaning that the optimisation needs to be carried out many times in order to be sure of exploring as much of the solution space as possible.

Chapter 6. Optimising for properties of the state

This added complexity is, as already discussed in Sec. 3.3.1, what generally makes CRAB a better approach in terms of results obtained and a more difficult one due to the computational overhead required for the optimisation.

In Fig. 6.2(a) we plot the results for the FO LCD, BPO and COLD for $N = 5$ spins, noting that unlike in the two spin case, the LCD approach for this set of operators no longer shows a significant speed up, with final state fidelities remaining low even at long times. In fact, for all plotted times, it shows barely a 1% improvement in fidelity over the bare Hamiltonian, which is not plotted as it would not be visible owing to it almost overlapping with the LCD line. The BPO approach, on the other hand, appears to perform better at longer times with a sharp increase in fidelity around what is likely a natural timescale for the system, where it dips below the COLD. The COLD approach, as before, performs really well at short driving times, where it is orders of magnitude better than either LCD or BPO. At longer times, this advantage wanes whether due to the optimisation process as multiple minima emerge in the cost function landscape or due to the fact that the local non-adiabatic effects are no longer the main source of losses.

Fig. 6.2(b) shows the results when using CRAB as well as COLD-CRAB in the same setting, with both performing far better at longer times than their naive BPO and COLD counterparts in plot (a) respectively, but remaining at similar levels of performance at short times, likely due to something more fundamental like the geometric speed limit [82]. This is definitely an argument for using something more sophisticated like COLD-CRAB as a general rule, as long as the computational resources are available for many and/or parallel optimisation instances.

While in Fig. 6.2 we only explore one optimisable parameter $N_k = 1$ and a system size of $N = 5$ spins, the resulting advantage of COLD over bare optimisation scales with system size and we find that, at least in the case of the naive pulse, increasing the number of parameters does not seem to make too much of a difference. These results and more discussion can be found in Appendix D.

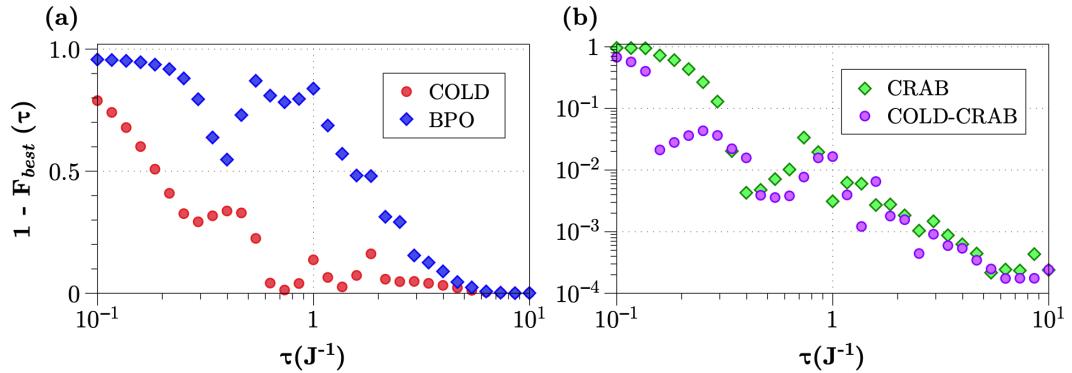


Figure 6.3: Optimisation of the constrained annealing protocol for the Ising model for $N = 5$ spins with a maximum amplitude limit on each term in the Hamiltonian of Eq. (6.15) of $X_0 = 10J_0$. (a) shows a comparison between BPO (blue diamonds). In (b) the comparison is between CRAB (green diamonds) and COLD-CRAB (purple circles). The plotted best results are obtained from 200 optimisations for each method. Figure reproduced from [1].

6.2.1 Restricting the driving amplitudes

While it is all well and good to talk about practical protocols implementing only local LCD operators with control drives that can be accessed by real experiments, one thing that we have so far failed to mention and which is not hard to observe from the form of the CD drive in Eq. 2.40, is that the amount of power required for the counterdiabatic pulse at short times scales with the speed of the changing Hamiltonian due to the λ coefficient. This means that while both LCD and COLD may lead to really high final state fidelities at very short driving times, they might also do this at the cost of impossibly high power requirements for the drives that implement the counterdiabatic component in either approach.

We find, (details in Fig. D.2 in Appendix D), that this is indeed what happens in the Ising chain case: as the total time of the evolution is reduced, the maximum amplitude reached by the LCD pulse increases, leading to two orders of magnitude in difference between the power requirements at $\tau = 10^{-3}J_0^{-1}$ and $\tau = 10^{-1}J_0^{-1}$. As one of the goals of COLD as a method is to be practically implementable, this result is inherently counterproductive.

In order to solve this issue, we can include penalty terms in the cost function used to

Chapter 6. Optimising for properties of the state

optimise the pulses for all of the different approaches: BPO, COLD, CRAB and COLD-CRAB. These penalty terms behave as constraints on the behaviour of the optimised pulse, and they may also include constraints on the LCD pulse that is included in COLD. If we take our original fidelity cost function from Eq. (3.14), we can modify it by including terms that are conditioned on the maximum amplitude of a given pulse:

$$C_F^{\text{const}}(\boldsymbol{h}, \boldsymbol{\beta}, \lambda) = 1 - F(\boldsymbol{h}, \boldsymbol{\beta}, \lambda) + \sum_m \Lambda_m(\boldsymbol{h}, \boldsymbol{\beta}, \lambda), \quad (6.22)$$

where $\Lambda_m(\boldsymbol{h}, \boldsymbol{\beta}, \lambda) = 0$ if the m^{th} constraint is satisfied and $\Lambda_m(\boldsymbol{h}, \boldsymbol{\beta}, \lambda) \gg \min(1 - F(\boldsymbol{h}, \boldsymbol{\beta}, \lambda))$ otherwise. We can implement such constraints for all of the driving amplitudes for all of the coefficients of each Hamiltonian, e.g. α , $f_{\text{opt}}^{N_k}$ and $f_{\text{CRAB}}^{N_k}$. In Fig. 6.3, this is done for the case where any of the drives exceeds the maximum amplitude of any of the bare Hamiltonian H_0 coefficients, which in our case is when $X(\lambda = 1) = X_0 = 10J_0$. We find that, while the resulting fidelities are greatly reduced in the BPO and COLD cases, the COLD approach outperforms the plain optimal control method. The far better results are obtained using CRAB and COLD-CRAB, which require far more computational resources but, as expected, allow a lot more flexibility within the constraints to obtain good final state fidelities.

The results presented here are supplemented further in Ch. 7, where we investigate how to use the ideas from Ch. 5 in order to implement CD-based cost functions to optimise COLD and BPO for the Ising spin chain. Additional plots for different system sizes and information on the variances between result outcomes in different optimisation instances are presented in Appendix D.

6.3 Transport in a synthetic lattice

Let us turn our sights to a completely different type of system for a moment and explore how COLD might perform. The efficient transfer of states between opposite ends of a lattice is an important protocol that could have future applications in the settings of quantum computation and simulation due to its promise of efficient transport of information [89]. This objective is often tackled in the setting of ultracold atoms in

Chapter 6. Optimising for properties of the state

optical lattices. While the problem can be tuned to be a single-particle system and the analytical solutions of the corresponding instantaneous Schrödinger equation are known [90, 91], the efficient evolution for state transfer is not straight-forward due to the states being largely delocalised across the lattice throughout the transfer.

In implementing CD, this delocalisation of states implies the requirement for the exact AGP operator to be highly delocalised too, which presents a practical difficulty. While such terms can be generated via the interactions of the atoms with cavity modes [92, 93] or from dipolar interactions [94, 95], the most tractable option remains that of approximate methods like LCD.

Recently, LCD was successfully applied to improve an adiabatic rapid passage (ARP) protocol for population transfer across a synthetic lattice [96]. In this realisation, population transfer was achieved in a synthetic tight-binding lattice of laser coupled atomic momentum states. We will consider the same problem as in [96] but with the improvement that can be gained by COLD.

The system is described by the Hamiltonian on N lattice sites

$$H_0(\mathbf{h}, \lambda) = - \sum_n^N J_n(\lambda)(c_n^\dagger c_{n+1} + H.c.) + \sum_n V_n(\lambda)c_n^\dagger c_n, \quad (6.23)$$

where $\mathbf{h} = \{J_n(\lambda), V_n(\lambda)\}_{n=1,\dots,N}$, $J_n(\lambda)$ is the λ -dependent tunnelling that describes the nearest-neighbour coupling, $V_n(\lambda)$ is the on-site energy offset with respect to neighbouring sites and $c_n^\dagger(c_n)$ is the creation(annihilation) operator on a given lattice site n . In the ARP protocol, the population gets moved from one end of the lattice to the other by linearly ramping the lattice from a positive tilt to a negative tilt via

$$\begin{aligned} J_n(\lambda) &= J_0(0.1 + \lambda) \\ V_n(\lambda) &= nV_0(1 - 2\lambda), \end{aligned} \quad (6.24)$$

where $V_0 = 4J_0$ is the initial site energy slope, J_0 is the characteristic tunnelling scale of the lattice and $\lambda(t) = \frac{t}{\tau}$ as previously.

In [96], the first order LCD is constructed by decomposing the tunneling into two

λ -dependent components:

$$J_n(\lambda) \rightarrow J_{n,\text{CD}}(\mathbf{h}, \lambda) e^{-i\phi_{n,\text{CD}}(\mathbf{h}, \lambda)}, \quad (6.25)$$

where

$$\begin{aligned} J_{n,\text{CD}}(\mathbf{h}, \lambda) &= \sqrt{J_n(\lambda)^2 + (\alpha_n(\mathbf{h}, \lambda)/\tau)^2}, \\ \phi_{n,\text{CD}}(\mathbf{h}, \lambda) &= \arctan\left(-\frac{J_n(\lambda)\tau}{\alpha_n(\mathbf{h}, \lambda)}\right), \end{aligned} \quad (6.26)$$

and the $\alpha_n(\lambda)$ terms correspond to the LCD coefficients. They can be found by solving a set of linear equations

$$\begin{aligned} &- 3(J_n J_{n+1})\alpha_{n+1} + (J_{n-1}^2 + 4J_n^2 + J_{n+1}^2)\alpha_n \\ &- 3(J_n J_{n-1})\alpha_{n-1} + (V_{n+1} - V_n)^2\alpha_n \\ &= -\partial_\lambda J_n(V_{n+1} - V_n). \end{aligned} \quad (6.27)$$

In order to implement COLD we once again have to include a control component into the Hamiltonian:

$$H_\beta(\mathbf{h}, \boldsymbol{\beta}, \lambda) = H_0(\mathbf{h}, \lambda) + \sum_n^N f_{\text{opt}}^{N_k}(\boldsymbol{\beta}, \lambda)(c_n^\dagger c_{n+1} + H.c.), \quad (6.28)$$

where the function $f_{\text{opt}}^{N_k} = \sum_k^{N_k} \beta_k(\lambda)$ is the same as in Eq. (6.19). This control pulse can be viewed as modifying the tunneling terms:

$$\begin{aligned} J_n(\lambda) &\rightarrow J_n^{\text{opt}}(\boldsymbol{\beta}, \lambda) = J_n(\lambda) + f_{\text{opt}}^{N_k}(\boldsymbol{\beta}, \lambda), \\ \Rightarrow J_{n,\text{CD}}(\boldsymbol{\beta}, \mathbf{h}, \lambda) &= \sqrt{J_n^{\text{opt}}(\boldsymbol{\beta}, \mathbf{h}, \lambda)^2 + (\alpha_n(\mathbf{h}, \boldsymbol{\beta}, \lambda)/\tau)^2} \\ \Rightarrow \phi_{n,\text{CD}}(\boldsymbol{\beta}, \mathbf{h}, \lambda) &= \arctan\left(-\frac{J_n^{\text{opt}}(\boldsymbol{\beta}, \lambda)\tau}{\alpha_n(\boldsymbol{\beta}, \mathbf{h}, \lambda)}\right), \end{aligned} \quad (6.29)$$

and the control functions are optimised using Powell's method as before by minimizing with respect to the fidelity of the final state, where the population has been fully transferred to the opposite lattice site.

Now that all of our ducks are in a row, we first consider a system size of $N = 7$

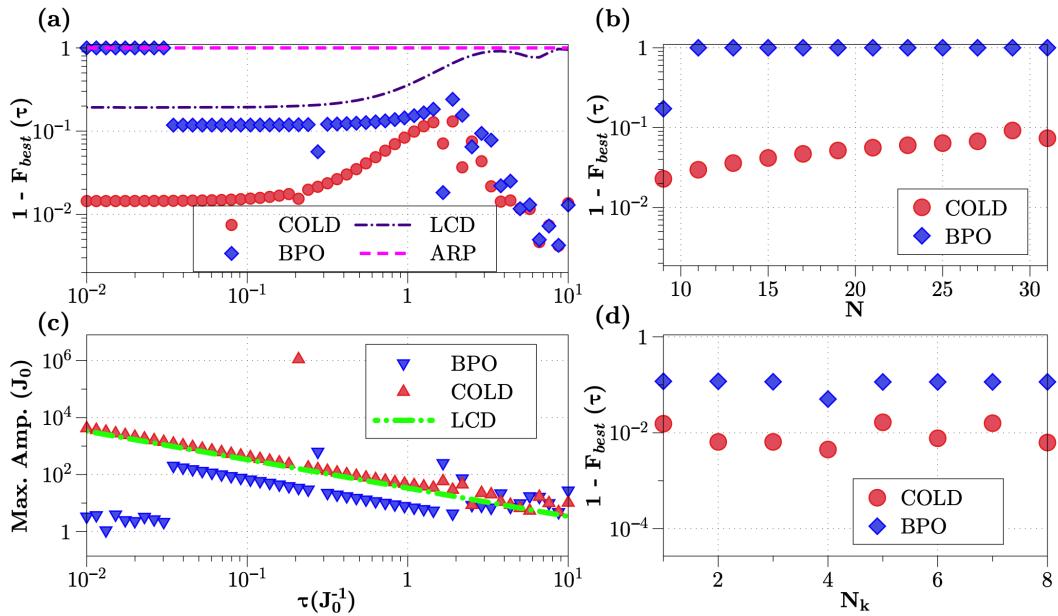


Figure 6.4: Optimisation of state transfer in a synthetic lattice. In (a) we compare the fidelities obtained via the bare ARP protocol (pink dashed line) and FO LCD implemented in [96] (purple dash-dot line) to BPO (blue diamonds) and COLD (red circles). (c) Maximum amplitude of the tunneling term at each driving time for LCD (green dash-dot line) as given by Eq. (6.25) as well as COLD (red triangles) which includes additional control parameters (Eq. (6.29)) and BPO (blue triangles) which omits the modifications due to CD but retains the control terms. In both (a) and (c) we simulate $N = 7$ lattice sites and use $N_k = 1$ parameter for optimisation of BPO and COLD. (b) Scaling of fidelities with increasing number of lattice sites (where $N_k = 1$) for both COLD (red circles) and BPO (blue diamonds). (d) does the same for the number of parameters while keeping $N = 7$. Note that both (b) and (d) are simulated for driving time $\tau = 0.5J^{-1}$ and the best fidelities are obtained across 500 optimisations. Figure reproduced from [1].

sites which was successfully experimentally probed in [96], where final state fidelities of 0.75 were achieved for $\tau = 1\text{ms}$ with a final tunnelling strength of $J/\hbar = 1/2\pi kHz$ (equivalent to $\tau \sim 1J_0^{-1}$ in our units). We initially confirm the breakdown of ARP in this setting for fast times, and the success of the LCD protocol at short times, as shown in Fig. 6.4 (a) and found in [96]. Implementing BPO on its own manages to enhance the achievable fidelities at intermediate times of $\tau > 0.03J^{-1}$. However, eventually, as observed in all scenarios in this work, BPO becomes stuck in the initial state at fast evolution times. Implementing the COLD protocol achieves an order of magnitude improvement in the fidelity over LCD. This is also plotted in Fig. 6.4(a).

Chapter 6. Optimising for properties of the state

It may be that COLD achieves this advantage by pumping power into the tunnelling term, as discussed in the Ising chain example, but we can see in Fig. 6.4(c) that the maximum amplitude of the tunnelling term tracks that of LCD. A key issue for experiments is the maximum amplitude achievable by a driving term and with this result we can stipulate that COLD is likely to be feasible in the same regimes as LCD in this synthetic lattice system, but with far higher resulting fidelities. There is single outlier at intermediate times as indicated by the single point peaking in maximum amplitude in Fig. 6.4(c), this is the exception to the rule, where the optimisation has found a marginally higher fidelity (see the corresponding point in Fig. 6.4(a)) by pumping in more power. Note

Furthermore, we explore the fidelities with increasing system size for both BPO and COLD in Fig. 6.4(b). While both protocols show a decreasing fidelity with increasing system size as expected, COLD does not suffer from getting stuck in the initial state, which is what happens in the BPO case as fidelities go to unity for larger systems in Fig. 6.4(b). This is the same mechanism as for the short driving times in Fig. 6.4(a). We also find, as plotted in Fig. 6.4(d), that increasing the number of optimisable parameters N_k in the control pulse does not contribute to an improvement in the results either for BPO or COLD. This means that, at least in this very simple control setting, there is no reason to expect that BPO will outperform COLD by simply adding more complexity to the control pulse.

It is important to acknowledge once again that it may be possible to achieve better results for both the control Hamiltonian and COLD with the use of more sophisticated optimal control methods like CRAB/COLD-CRAB or a global optimiser instead of Powell’s method. Any of these methods might prove to be better, but they are also far more computationally intensive and the results presented already show significant improvements over LCD or the bare Hamiltonian. In the case where such a protocol is to be implemented in practice, it would be advantageous to explore more refined control methods than those presented here.

6.4 Preparing GHZ states in a system of frustrated spins

Multipartite entanglement is a powerful resource for quantum computing and more broadly in quantum technologies, offering unique capabilities for information processing, secure communication, high-precision measurements, and understanding the foundations of quantum mechanics [Reminder: does this need citations?](#). An example of such highly entangled states is the GHZ (Greenberger–Horne–Zeilinger) state [97] on $N > 1$ spins:

$$|\text{GHZ}\rangle = \frac{1}{\sqrt{2}}(|0\rangle^{\otimes N} + |1\rangle^{\otimes N}), \quad (6.30)$$

here written in the 0, 1 qubit basis.

It is possible to prepare such states in a system of frustrated spins (see Fig. 6.5(a)) for odd $N > 1$ via an annealing protocol. The Hamiltonian describing such a system is

$$H_0(\mathbf{h}, \lambda) = J(\lambda) \left(\sum_j^{N-1} \sigma_j^z \sigma_{j+1}^z + \sum_j^{N-2} \sigma_j^z \sigma_{j+2}^z \right) + h(\lambda) \left(\sum_j^N (\sigma_j^z + \sigma_j^x) \right). \quad (6.31)$$

where $\mathbf{h} = \{J(\lambda), h(\lambda)\}$ with $\lambda = \frac{t}{\tau}$, $J(\lambda) = -J_0$ and

$$h(\lambda) = -h_0 \left(1 - \sin^2 \left(\frac{\pi}{2} \sin^2 \left(\frac{\pi}{2} \lambda \right) \right) \right). \quad (6.32)$$

The parameters used for the results in this section are $J_0 = 1$ and $h_0 = 10J_0$, meaning the spins start in close to the $|\downarrow^{\otimes N}\rangle$ state.

In this case, we will apply both FO LCD and SO LCD ansätze. The former is the same one as used previously for the Ising chain case, Eq. (6.17), while the latter will consist of additional operators between next-nearest-neighbour spins to reflect the geometry of the bare Hamiltonian:

$$\begin{aligned} \mathbb{A}_\lambda^{(2)}(\mathbf{h}, \lambda) &= \gamma(\mathbf{h}, \lambda) \left(\sum_j^{N-1} \sigma_j^x \sigma_{j+1}^y + \sum_j^{N-2} \sigma_j^y \sigma_{j+2}^x \right) \\ &\quad + \zeta(\mathbf{h}, \lambda) \left(\sum_j^{N-1} \sigma_j^z \sigma_{j+1}^y + \sum_j^{N-2} \sigma_j^y \sigma_{j+2}^z \right). \end{aligned} \quad (6.33)$$

Chapter 6. Optimising for properties of the state

As H_0 is simply the Ising Hamiltonian with added couplings between spins j and $j+2$, its LCD coefficients can be found by using the coupled equations in Appendix C and setting the unused operator coefficients to 0.

What remains is to construct the optimal control pulse, which in this case will be done using the GRAPE method from Sec. 3.3.2. The control Hamiltonian is

$$H_\beta(\boldsymbol{\beta}, \mathbf{h}, \lambda) = H_0(\mathbf{h}, \lambda) + f_{\text{GRAPE}}^{N_k}(\boldsymbol{\beta}, \lambda) \sum_j^N \sigma_j^z, \quad (6.34)$$

where the pulse $f_{\text{GRAPE}}^{N_k}$ is comprised of N_k time slices $\Delta\lambda = \lambda_{k+1} - \lambda_k$, such that $N_k \times \Delta\lambda = 1$. During each slice λ_k , a piecewise constant control amplitude $\beta_k(\lambda_k) = c_k$, is applied to the control system, with c_k denoting the optimisable parameter which represents the amplitude of the pulse in the interval $[\lambda_k, \lambda_{k+1}]$. Thus, the pulse comprises of N_k optimisable parameters for N_k time slices. In order to enforce the constraints that the control pulse vanish at the beginning and end of the protocol and to make the pulse more smooth, we also apply a shaping function, which acts during each time interval $[\lambda_k, \lambda_{k+1}]$ as

$$f_{\text{shape}}(\beta_k, \lambda_k) = c_k \tanh(\kappa\theta(\lambda_k)) \tanh(-\kappa\theta(\lambda_k - \tau)), \quad (6.35)$$

with $\theta(\lambda) = \sin \frac{\pi}{2}\lambda$ and $\kappa = 30$ an offset parameter. As discussed in Sec. 4.2, we use spline interpolation to calculate the derivatives of the control drive when they are required to obtain the LCD drives. The resulting function requires more parameters than the chopped basis we chose to use in previous examples, but it also allows for more flexibility in the final shape of the drive. Furthermore, due the increased number of parameters and search space, instead of Powell optimisation as in previous examples we choose to instead implement dual annealing, first presented in detail in Sec. 3.1.3.3. Dual annealing is a global optimiser and while computationally more costly, is far better in the case of a complex parameter space with multiple minima, which is what may be expected in this case (see Fig. 7.5 later in the thesis). In this case, instead of referring to the optimised control Hamiltonian of Eq. (6.34) as BPO, we will instead dub it ‘bare

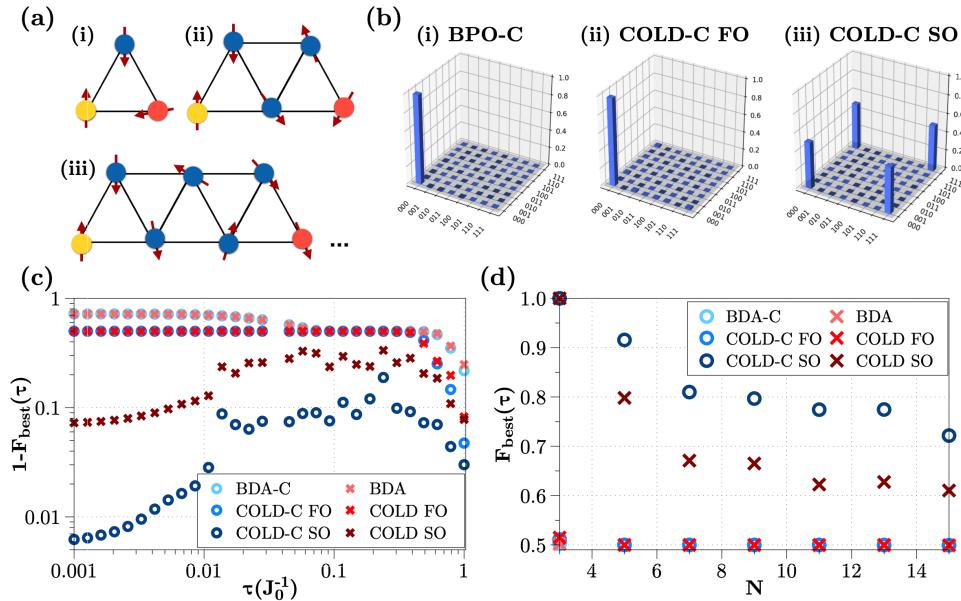


Figure 6.5: GHZ state preparation in systems of frustrated spins. Spins are arranged in triangular formations as depicted in (a) for (i) 3, (ii) 5 and (iii) 7 spins, with spins on the vertices and edges representing couplings. In the case of corner optimisation, three separate optimisable drives are applied: one for the yellow corner spin, one for the red corner spin and a third drive for all of the blue spins in-between. (b) Density matrix plots of the final state of a 3 spin triangle after an evolution time $\tau = 0.1J_0^{-1}$ when optimised using (i) BDA-C, (ii) FO COLD-C and (iii) SO COLD-C. (c) Final fidelities of the GHZ state on 5 spins for optimised global drive (red crosses) and locally driven corner spins (blue rings). (d) Final fidelities at driving time $\tau = 0.1J_0^{-1}$ for different systems sizes N . In the global case we use $N_k = 10$ and in the corner case the total parameters are $3 \times N_k = 30$. The plots are for best results of 5 optimisations for each data point and the dual-annealing search space was bounded in the range $[-50, 50]$ for all parameters. Figure reproduced from [1].

dual-annealing' or BDA.

As well as implementing the GRAPE pulse for all spins as in Eq. 6.34, as this is a more interesting and complex system than those encountered previously, we will include a separate protocol where three separate control pulses are used: one for each corner spin (see Fig. 6.5(a)) and one for all the spins in-between. The reason for this choice of pulses is because it is often easier to independently address the edges of a lattice in practical implementations of such protocols, rather than attempting to have local control of all of the spins individually. We will include a '-C' suffix to each method in order to differentiate the corner approach from that of a global control pulse. The

Chapter 6. Optimising for properties of the state

resulting control Hamiltonian will thus have $3 \times N_k$ optimisable parameters, which is a very large search space, making the optimisation process very computationally expensive when compared to all of the previous examples discussed in this Chapter.

In Fig. 6.5(c) we plot the resulting final state fidelities at different total driving times τ for a 5 spin frustrated system. We implement BDA, FO and SO COLD as well as their corner-optimised versions while optimising the pulse parameters for final state fidelity with respect to the GHZ state of Eq. (6.30). We observe that FO COLD is not particularly effective at short driving times and does not move the system out of its initial state (see the density matrix plots in (b)), regardless of whether or not separate control is applied to the corner spins. This is very likely due to the fact that the σ^y terms making up FO LCD are only a small contribution to the full counterdiabatic drive and thus we need to look to higher-order ansätze to see improvements. Notably, the bare control protocol BDA does not fare any better, refusing to budge from the initial state at very small τ . SO COLD, on the other hand, shows a five-fold improvement over the first order when a global optimisable drive is applied and up to two orders of magnitude improvement when the corner spins are driven separately at short times ($\tau = 0.001J_0^{-1}$). We run optimisations for larger systems at time $\tau = 0.1J_0^{-1}$ and find that this advantage is retained even with increasing system size, as plotted in Fig. 6.5(d). This is a large improvement over a recent result presented in [98], where optimal control was used to directly determine SO LCD coefficients for preparing the GHZ state on an Ising chain rather than creating a control Hamiltonian. At 10 spins the final state fidelity for $\tau = 1J_0^{-1}$ obtained in their paper was 0.18, while we reach a fidelity of 0.72 for 15 spins when using corner optimisation at $\tau = 0.1J_0^{-1}$.

6.4.1 Tripartite GHZ entanglement

While throughout this chapter we have focused on using final state fidelity as a cost function for the optimisation, in the case of GHZ state preparation, the goal is often the maximal entanglement of such states. Therefore, we can imagine using a cost function which maximises some entanglement metric like that in Eq. (3.16), rather than the final state fidelity. This can be done in cases where several maximally entangled states are

Chapter 6. Optimising for properties of the state

targeted, or where a single target state maximises entanglement. In the latter case, the advantage may lie in the fact that entanglement as a metric might lead to a smoother and more convex cost function landscape.

The GHZ state exhibits a particular type of entanglement: when one of the subsystems is measured, the rest are no longer entangled and collapse into a product state. This is different to the other canonical type of multipartite entanglement exhibited by the W state [99], which for 3 spins can be written as:

$$|W\rangle = \frac{1}{\sqrt{3}}(|001\rangle + |010\rangle + |100\rangle). \quad (6.36)$$

While measuring the entanglement of a multipartite system is not quite as simple 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. [100], which is maximised when a three spin system exhibits maximal GHZ-type entanglement and minimised for W-type entanglement and product states. The three-tangle is a very efficient metric and can be expressed as

$$\begin{aligned} T_3(|\psi\rangle) &= 4|d_1 - 2d_2 + 4d_3|, \\ d_1 &= c_{000}^2 c_{111}^2 + c_{001}^2 c_{110}^2 + c_{010}^2 c_{101}^2 + c_{011}^2 c_{100}^2, \\ d_2 &= c_{000} c_{001} c_{110} c_{111} + c_{000} c_{010} c_{101} c_{111} + c_{000} c_{011} c_{100} c_{111} \\ &\quad + c_{001} c_{010} c_{101} c_{110} + c_{001} c_{011} c_{100} c_{110} + c_{010} c_{011} c_{100} c_{101} \\ d_3 &= c_{000} c_{110} c_{101} c_{011} + c_{100} c_{010} c_{001} c_{111}, \end{aligned} \quad (6.37)$$

where c_{ijk} represents the complex coefficient of the state $|ijk\rangle$ of the three spin state $|\psi\rangle$.

In Fig. 6.6(a) we investigate the values of T_3 obtained when optimising for state fidelity in a three spin system like that of Fig. 6.5(a)(i) with only a global control pulse and dual-annealing optimisation. We find that the value of the three tangle and hence the amount of entanglement in the system begins to increase prior to any noticeable improvement in fidelity in the case of BDA and FO COLD, while, as expected, given the much higher fidelities obtained when using SO COLD, the entanglement is max-

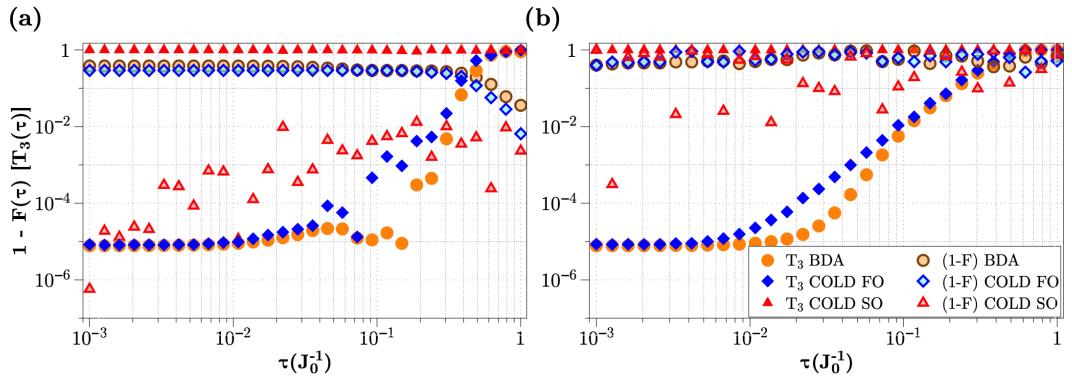


Figure 6.6: Plots of T_3 (Eq. (6.37), solid shapes) and fidelity ($1 - F$) with respect to the GHZ state (outlined shapes) of the final state prepared after optimising a GRAPE pulse with $N_k = 6$ parameters for each value of τ . In (a) optimisation is performed using final state fidelity as the cost function and in (b) the cost function is the three tangle from Eq. (6.38). In both cases, the result is from a single optimisation for each data point and the dual-annealing search space bounds are $[-50, 50]$ for all optimisable parameters.

imised for even for very small τ . This is an interesting result, as it indicates that SO counterdiabatic operators are required to be able to speed up entanglement generation. Even when FO terms are applied, it long evolution times to generate entanglement.

Fig. 6.6(b) on the other hand, shows the results when optimising solely for the three tangle. The cost function is

$$C_{T_3}(\boldsymbol{\beta}, \tau) = 1 - T_3(|\psi_f(\boldsymbol{\beta}, \tau)\rangle), \quad (6.38)$$

where $|\psi_f(\boldsymbol{\beta}, \tau)\rangle$ is the final state obtained throughout the evolution during time τ and with optimal controls $\boldsymbol{\beta}$. What we find is that even when maximising for entanglement, the results for final T_3 values are not very different to those in (a) when optimising for final state fidelity. This is again an indication that neither FO terms nor the plain control Hamiltonian of Eq. (6.34) are enough to generate entanglement quickly and that SO terms are necessary for this purpose. The much lower fidelities in plot (b) are unsurprising, as the three-tangle is a measure of GHZ-type entanglement, which is maximised for several states including the orthogonal state to that presented in

Chapter 6. Optimising for properties of the state

Eq. (6.30). This means that maximising for entanglement in this case will not always lead to final states that have high fidelity with respect to the canonical GHZ state presented at the start of this section.

Chapter 7

Higher order AGP as a cost function

Hello

Hi

In Ch. 5 we discussed the idea of using the AGP operator and its approximations in order to construct cost functions for the optimisation of Hamiltonian paths in parameter space. There are several reasons why one might expect this to be a good idea: for example, a path in the Hamiltonian parameter space which minimises the AGP norm should, in principle, also minimise the non-adiabatic effects experienced by the system when it is driven along that path. Furthermore, such cost functions should be very efficient to compute once they are written in the correct functional form, giving them an advantage over the fidelity cost function that we used in most examples of Ch.6, which suffers from increasing complexity and inefficiency with growing system sizes due to requiring access to the system wavefunction.

In this chapter we will motivate the idea of AGP-based cost functions with numerical results, investigating two types of cost functions presented in Ch. 5: minimisation of absolute integrals of the LCD coefficients and minimisation of their maximal amplitudes. We will do this for three different example systems which we covered in the previous chapter: two-spin annealing in Sec. 7.1, the Ising spin chain in Sec. 7.2 and finally the preparation of maximally entangled GHZ states in systems of frustrated spins

Chapter 7. Higher order AGP as a cost function

in Sec. 7.3. In the last example, we will find a situation where this new approach might not be optimal and discuss the reasons behind why, with the goal of understanding when such an approach is or isn't effective.

7.1 Return to two-spin annealing

In the previous chapter in Sec. 6.1 we investigated the COLD protocol in the case of a two-spin annealing protocol described by the Hamiltonian given by Eq. (6.1), where the system starts close to the state $|\uparrow\uparrow\rangle$ and is driven towards a superposition of all the symmetric states. We will return to this simple example in order to illustrate how the integral and maximum amplitude cost functions from Sec. 5, given by Eq. (5.1) and Eq. (5.2) respectively, behave when used to optimise the Hamiltonian path in parameter space for COLD.

The key difference between what was done in Ch. 6 and what we wish to demonstrate here is the use of different cost functions in the optimisation of the parameters c_k from Eq. (6.12). As discussed in Ch. 5, once we have the AGP or LCD operators expressed as functions of the control Hamiltonian, including a dependence on the control functions $\beta_k \in \boldsymbol{\beta}$, they can be used to construct cost functions that can be evaluated very efficiently, regardless of the scale or complexity of the driven system. This is important, as while the fidelity cost function C_F from Eq. (3.14) that we used in the previous chapter is very effective, given that it evaluates exactly how close we are to the goal of the optimisation, its efficiency scales very poorly with increasing system size.

In this example, we will use the same control Hamiltonian as that the two-spin example from the previous chapter given by Eq. (6.12) as well as the same FO and SO operator ansätze for the LCD operators (Eq. (6.3) and Eq. (6.8) respectively). We found previously that the coefficient $\alpha(\boldsymbol{\beta}, \mathbf{h}, \lambda)$ which drives the FO LCD terms and the coefficients $\gamma(\boldsymbol{\beta}, \mathbf{h}, \lambda)$ and $\zeta(\boldsymbol{\beta}, \mathbf{h}, \lambda)$ driving the SO LCD terms can be found by solving the coupled set of equations given in Eq. (6.9). We noted that these three coefficients and the operators they drive are enough to describe the exact CD pulse for any given parameters $(\boldsymbol{\beta}, \mathbf{h}, \lambda)$, since the two-spin system is a simple example.

We know that applying the exact CD pulse made up of all of the FO and SO LCD

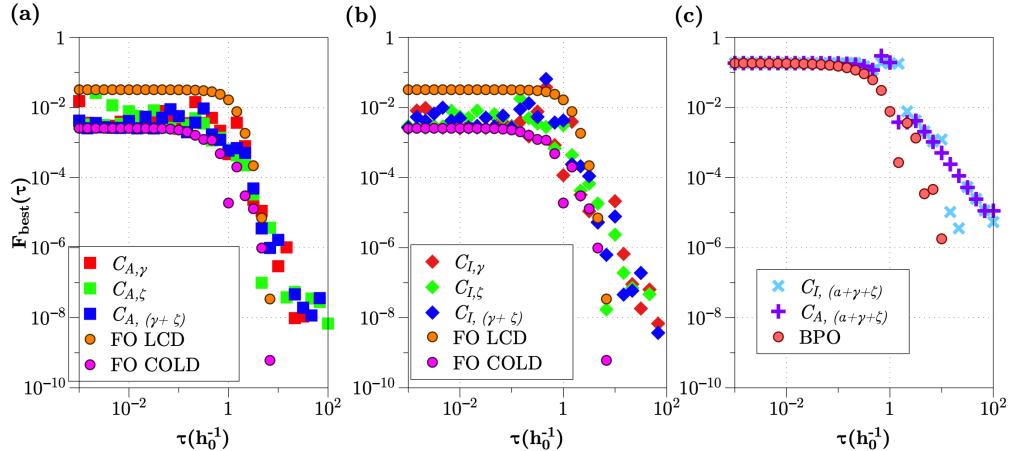


Figure 7.1: Two-spin annealing plots of fidelity versus driving time when optimising using different cost functions. In (a)-(b), we show the results for FO LCD (orange circles) and FO COLD (pink circles) from Fig. 6.1 for comparison. We do the same in (c) for BPO (red circles). Then, in (a) we plot the resulting final state fidelities at different driving times τ obtained when applying FO COLD to control Hamiltonians from Eq. (6.12) with parameters optimised using the maximum amplitude cost function C_A . The results are shown when optimising for γ coefficients (red squares, $C_{A,\gamma}$), ζ coefficients (green squares, $C_{A,\zeta}$) and their sum (blue squares, $C_{A,(\gamma+\zeta)}$). The same is done in (b) for the integral cost function C_I , where we plot results in the case of optimising γ coefficients (red diamonds, $C_{I,\gamma}$), ζ coefficients (green diamonds, $C_{I,\zeta}$) and their sum (blue diamonds, $C_{I,(\gamma+\zeta)}$). In (c), we plot the resulting final state fidelities when BPO is applied with a control pulse optimised using $C_{I,(\alpha+\gamma+\zeta)}$ (light blue crosses) and $C_{A,(\alpha+\gamma+\zeta)}$ (purple pluses). In all cases, we use $N_k = 1$. The optimisation is done 10 times for each data point with the best final fidelity plotted. We use the Powell optimisation method from Sec. 3.1.3.2 for the minimisation.

terms returns unit fidelity regardless of driving time or any optimisable parameters, as shown in Fig. 6.1. We also found, plotted in the same Figure, that applying FO LCD to the problem without any control pulse performed worse than applying COLD with FO terms, where a control pulse is included and the control parameters are optimised for final state fidelity. In this case, we will use a series of cost functions constructed in a similar manner to Eq. (5.1) and Eq. (5.2) in order to optimise the COLD control pulse without computing the final state fidelity until after the optimisation is done, in order to check how well the optimisation has performed. First, we define the cost functions

which use the magnitudes of the coefficient integrals as:

$$\begin{aligned} C_{I,\gamma}(\tau, \boldsymbol{\beta}) &= \int_0^\tau dt' |\gamma(\lambda(t'), \mathbf{h}, \boldsymbol{\beta})|, \\ C_{I,\zeta}(\tau, \boldsymbol{\beta}) &= \int_0^\tau dt' |\zeta(\lambda(t'), \mathbf{h}, \boldsymbol{\beta})|, \\ C_{I,(\gamma+\zeta)}(\tau, \boldsymbol{\beta}) &= \int_0^\tau dt' (|\gamma(\lambda(t'), \mathbf{h}, \boldsymbol{\beta})| + |\zeta(\lambda(t'), \mathbf{h}, \boldsymbol{\beta})|), \\ C_{I,(\alpha+\gamma+\zeta)}(\tau, \boldsymbol{\beta}) &= \int_0^\tau dt' (|\alpha(\lambda(t'), \mathbf{h}, \boldsymbol{\beta})| + |\gamma(\lambda(t'), \mathbf{h}, \boldsymbol{\beta})| + |\zeta(\lambda(t'), \mathbf{h}, \boldsymbol{\beta})|), \end{aligned} \quad (7.1)$$

where the subscript I denotes an integral-based cost function and the letters γ and ζ simply refer to which pulse coefficient is being minimised for the optimisation process. We then do the same for the maximum amplitude of the pulses, using the subscript A to denote ‘amplitude’:

$$\begin{aligned} C_{A,\gamma}(\tau, \boldsymbol{\beta}) &= \max_{t' \in [0, \tau]} |\gamma(\lambda(t'), \mathbf{h}, \boldsymbol{\beta})|, \\ C_{A,\zeta}(\tau, \boldsymbol{\beta}) &= \max_{t' \in [0, \tau]} |\zeta(\lambda(t'), \mathbf{h}, \boldsymbol{\beta})|, \\ C_{A,(\gamma+\zeta)}(\tau, \boldsymbol{\beta}) &= \max_{t' \in [0, \tau]} (|\gamma(\lambda(t'), \mathbf{h}, \boldsymbol{\beta})| + |\zeta(\lambda(t'), \mathbf{h}, \boldsymbol{\beta})|), \\ C_{A,(\alpha+\gamma+\zeta)}(\tau, \boldsymbol{\beta}) &= \max_{t' \in [0, \tau]} (|\alpha(\lambda(t'), \mathbf{h}, \boldsymbol{\beta})| + |\gamma(\lambda(t'), \mathbf{h}, \boldsymbol{\beta})| + |\zeta(\lambda(t'), \mathbf{h}, \boldsymbol{\beta})|), \end{aligned} \quad (7.2)$$

which are used to optimise the parameters $\boldsymbol{\beta}$ by minimising the maximum amplitude of the given LCD coefficients.

The results of optimising for the SO LCD coefficients γ and ζ are plotted in Fig. 7.1(a) and (b). In (a) we show the results of using $C_{A,\gamma}$, $C_{A,\zeta}$ and $C_{A,(\gamma+\zeta)}$ as cost functions for optimising a FO COLD pulse, with FO LCD and FO COLD (optimised using fidelity) from Fig. 6.1 plotted for comparison. We do the same for the respective integral cost functions in (b). The results indicate that while optimising for final state fidelity, not unexpectedly, shows better final state fidelities when COLD is applied, the amplitude and integral cost functions still generally perform better in producing a state close to the target ground state than a naive application of FO LCD with no optimal control. This is a positive result, as it implies that there is a correlation between properties of the LCD coefficients and the final state fidelity in designing

Chapter 7. Higher order AGP as a cost function

optimal control pulses. This is exemplified further in plot (c) of the figure, where we optimise the pulse for a case when no LCD is applied, using a minimisation of the exact CD pulse comprised of all three LCD coefficients. At short evolution times, the optimisation performs as well as the fidelity cost function, while at longer times it begins to lag a little, with a few outliers appearing potentially due to local minima in the cost function landscape. This is to be expected, as minimisation of the exact CD pulse should be equivalent to the minimisation of non-adiabatic effects experienced by the system, and at very short driving times these will probably dominate the loss of final stat fidelity, while at longer times there may be several paths in parameter space that are similarly effective.

In order to better understand the relationship between the LCD coefficients and the value of the final state fidelity with respect to the target, in Fig. 7.2(a-c) we plot the cost function landscapes for fidelity (C_F) at $\tau = 0.1h_0^{-1}$ for two parameters c_1 and $c - 2$ in the range $[-10, 10]$. In (a) we plot the value of C_F when FO COLD is applied, which corresponds to the C_F landscape for the orange circle plots in Fig. 7.1(a-b). Then, in (b) we plot the values of C_F when COLD with only the ζ terms $\sigma_1^y \sigma_2^z + \sigma_1^z \sigma_2^y$ is applied. Finally, in (c), we plot C_F for COLD with both of the SO terms γ and ζ applied. We then do the same for $C_{I,\alpha}$ in (d), $C_{I,\zeta}$ in (e) and $C_{I,(\gamma+\zeta)}$ in (f). We then do the same for the respective maximum amplitude cost functions. The α coefficients for (a), (d) and (g) are obtained by solving the coupled equations of Eq. (6.9) with γ and ζ set to 0, while for the plots in (b),(e) and (h) we set α to 0, and find that the γ pulse turns out to be 0 for all values of β and λ , hence retaining only values of ζ . In (c),(f) and (i) we solve for the full CD pulse with all coefficients but only plot the SO components. What we find is that while there appears to be some relationship between the maximum and minimum of the AGP cost functions and the final state fidelities, it is not clear cut. Certainly, where the final state fidelity $F(\tau)$ is maximised in (a) (i.e. when the plot shows a minimum value as $C_F = 1 - F$), the SO LCD plots (e-f) and (h-i) have small values. However, it appears as though the SO components are also small for values of c_1 and c_2 which lead to very bad final state fidelities in (a). It is also clear that maximising fidelity for FO COLD is not necessarily equivalent to maximising the FO

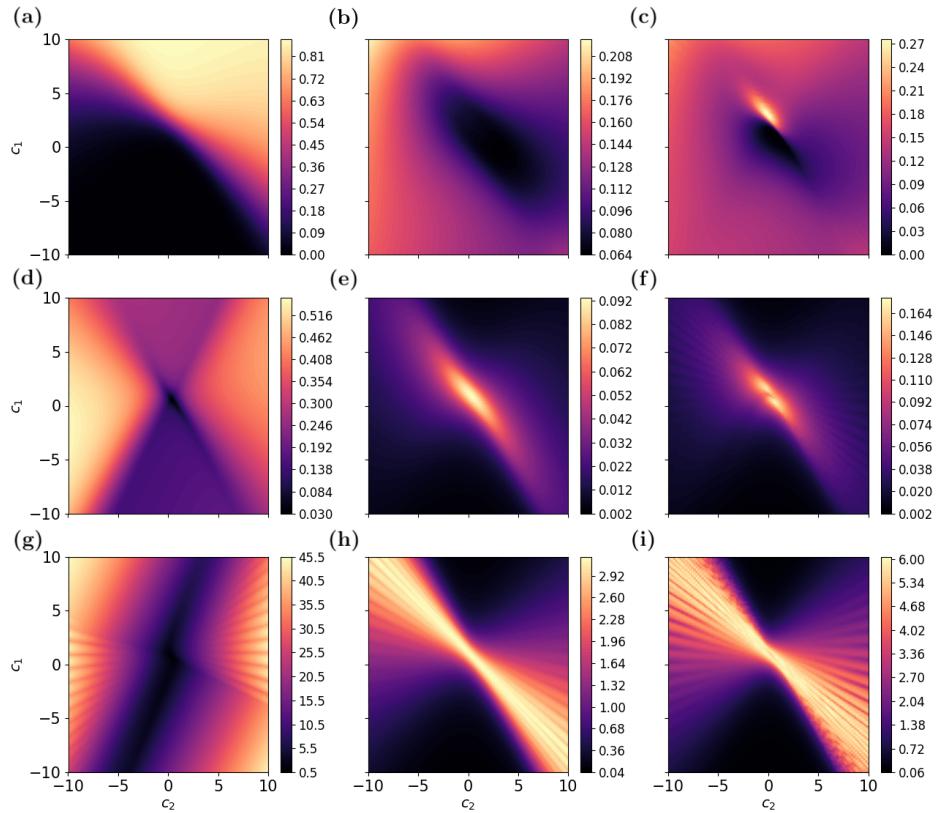


Figure 7.2: Contour plots of fidelity and AGP cost function landscapes for two parameters $c_1, c_2 \in [-10, 10]$ for two spin annealing, as discussed in the text. All plots are for total evolution time $\tau = 0.1h_0^{-1}$. (a-c) show plots of fidelity cost function C_F values when (a) FO COLD is applied (α terms), (b) SO COLD ζ terms are applied and (c) both SO COLD terms γ and ζ are applied as discussed in the text. (d-f) show plots of the integral cost function C_I values for (d) the coefficient α , $C_{I,\alpha}$ (e) the coefficient ζ , $C_{I,\zeta}$ and (f) sum of the coefficients γ and ζ , $C_{I,\gamma+\zeta}$. (g-i) do the same for C_A , where (g) shows a plot of $C_{A,\alpha}$, (h) shows $C_{A,\zeta}$ and (i) is a plot of $C_{A,\gamma+\zeta}$.

LCD coefficient. What these plots are intended to illustrate is that while there appears to be some relationship between the various approximations of the AGP and the final state fidelity of the system when COLD of various orders is applied, this relationship need not be clear cut.

We note that there is no contour plot for C_F in the case of BPO nor plots of the integral or amplitude functions for the exact CD pulse comprised of all of the LCD

Chapter 7. Higher order AGP as a cost function

coefficients and that is because the optimal parameter values of c_1 and c_2 were very large (of the order of $1 \times 10^3 h_0^{-1}$) and varied in the results of Fig. 7.1(c), making it difficult to capture the relevant cost function landscapes visually. The fact that the COLD optimal control pulses require quite low amplitudes even at short driving times could, in fact, also be considered an advantage of the method.

While in this simple example it may be more favorable to implement the fidelity cost function C_F given that for two spins it is reasonably efficient to compute the state evolution, in more complex systems this is no longer the case. Iterative optimisation procedures like Powell's method (Sec. 3.1.3.2), which we have been using, may require hundreds or thousands of cost function evaluations for a single optimisation procedure. The fact that we get results which are comparable to C_F while using a set of cost functions which become exponentially more efficient to implement as the system size grows is something that can become very useful in such more complex cases.

7.2 Return to the Ising spin chain

As in the previous section, here we revisit a system that was already explored in the previous chapter: the Ising spin chain of Sec. 6.2. As in the two-spin case, we are interested in retaining the same parameters as those explored for COLD with the fidelity cost function, changing only the optimisation landscape via the integral and amplitude cost functions outlined in Eq. (7.1) and Eq. (7.2) respectively. Thus, we use the same bare Hamiltonian and parameters (Eq. (C.1)) along with the Powell optimal control pulse from Eq. (6.19), parameterised by the control functions $\beta_k \in \boldsymbol{\beta}$ and by constant control parameters c_k . The FO LCD ansatz is still the same as in Eq. (6.17), which is a set of local σ^y operators on each spin and the SO operators are taken to be all of the next-neighbour, two-body Pauli terms on N spins, expressed as

$$\begin{aligned} \mathbb{A}_\lambda^{(2)}(\boldsymbol{\beta}, \mathbf{h}, \lambda) = & \sum_j^{N-1} \gamma(\boldsymbol{\beta}, \mathbf{h}, \lambda) (\sigma_j^x \sigma_{j+1}^y + \sigma_j^y \sigma_{j+1}^x) \\ & + \sum_j^{N-1} \zeta(\boldsymbol{\beta}, \mathbf{h}, \lambda) (\sigma_j^z \sigma_{j+1}^y + \sigma_j^y \sigma_{j+1}^z), \end{aligned} \quad (7.3)$$

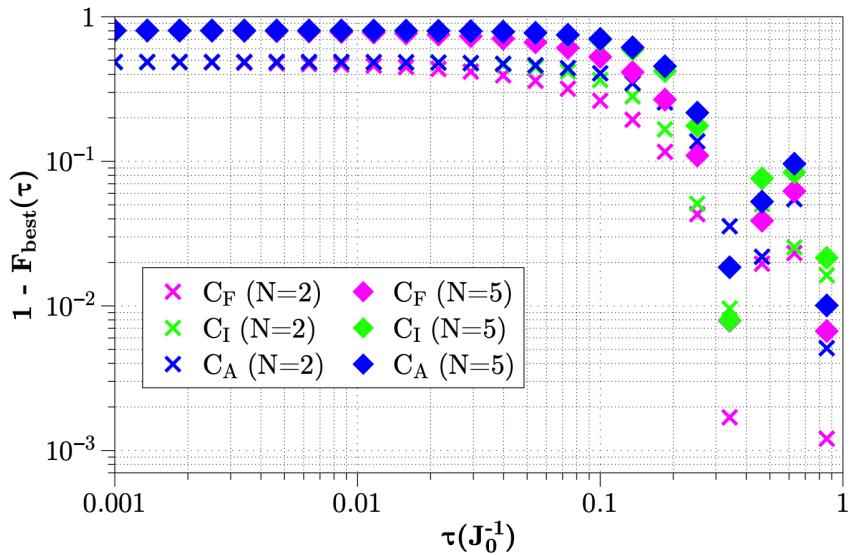


Figure 7.3: Plot of final state fidelities obtained when optimising for the fidelity cost function C_F (pink), the integral cost function for FO and SO LCD pulses $C_{I,(\alpha+\gamma+\zeta)}$ (green) as well as the maximum amplitude cost function $C_{A,(\alpha+\gamma+\zeta)}$ (blue). We plot results for two spins (crosses) and five spins (diamonds), optimising separately for both. Results are plotted for $N_k = 1$ and different total driving times τ . Optimisation is performed 10 times for each data point and the lowest obtained value for the cost function is used to compute the fidelity in the case of C_A and C_I .

where γ and ζ are the SO LCD coefficients as in the previous example. In this section we use the same integral and amplitude cost functions as in Eq. (7.1) and Eq. (7.2), with the coefficients α , γ and ζ as presented above. These can be solved for by using the results presented in Appendix C, as discussed in Sec. 6.2.

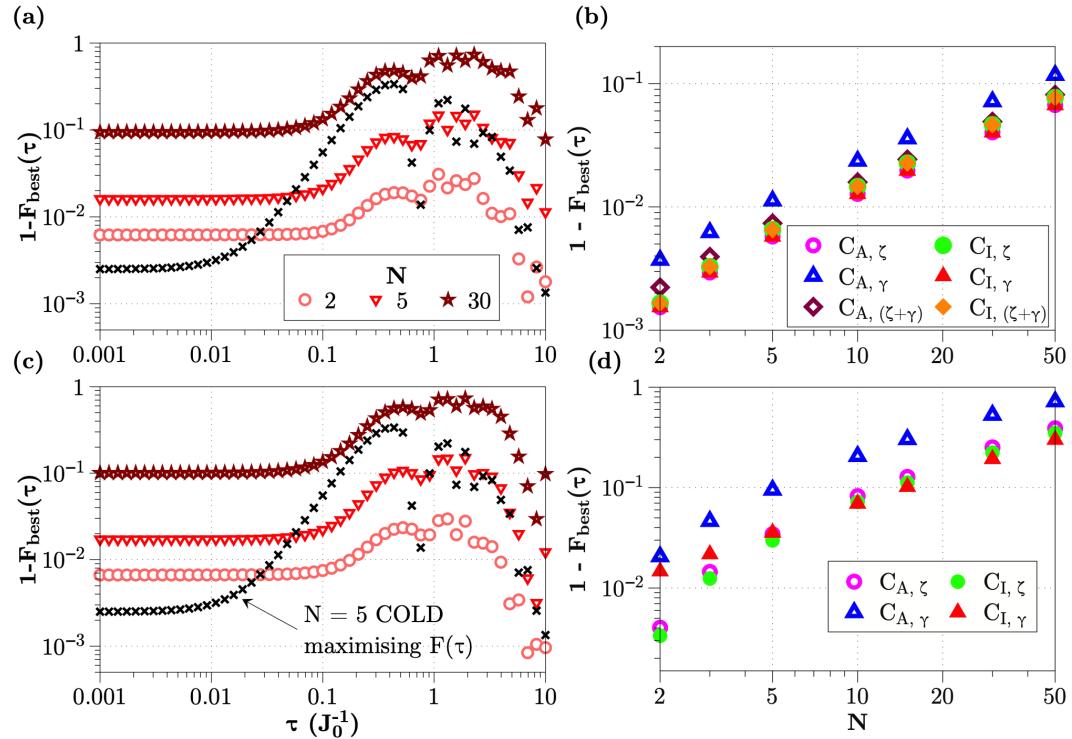
The main difference between the two spin example and this one, in particular when considering using AGP constructed cost functions, is that for increasing spin chain lengths we can no longer expect to have access to the exact CD pulse, as the non-adiabatic effects may become delocalised quickly throughout the chain. Even if the effects of the delocalised AGP operators are small, they are not necessarily guaranteed to be non-zero. Thus, we are now operating in a setting where we might not have all of the information about the non-adiabatic effects on the system and must instead contend with LCD approximations of the exact counterdiabatic pulse explicitly.

With this in mind, the first thing we explore is whether or not we can use the

Chapter 7. Higher order AGP as a cost function

FO and SO LCD coefficients in the same vein as in Fig. 7.1(c) to optimise the BPO pulse. We plot the results in Fig. 7.3 for the cases of $N = 2$ spins and $N = 5$ spins, using C_F , $C_{A,(\alpha+\gamma+\zeta)}$ and $C_{I,(\alpha+\gamma+\zeta)}$. We expect that in the two-spin case we will get a similar result as in Fig. 7.1(c), given that we should be minimising the exact CD pulse in the case of two spins when using both FO and SO in the cost function and this appears to be the case: at short evolution times, the results of the final state fidelity match up regardless of which cost function is used, while at longer times C_F begins to perform better, although all cost functions show a similar pattern in final state fidelity scaling with τ . However, in the five spin case, we do not have an explicit reason to expect a similar behaviour unless higher order LCD does not play a big part in the non-adiabatic effects experienced by the system. It turns out, in fact, that this is the case: the behaviour for five spins is comparable to that of two spins: at short times the cost functions are equally as effective, with differences appearing only at longer times.

The other case that we can explore, then, to improve our results, is to see if we could implement FO or partial SO COLD while minimising the other LCD term coefficients. This is presented in Fig. 7.4, where in (a-b) we implement COLD with FO LCD coefficients while minimising the integral cost function $C_{I,(\zeta)}$ in (a) and the amplitude cost function $C_{A,(\zeta)}$ in (c), both for system sizes of $N = 2, 5, 30$. We also plot the results from Fig. 6.2(a) which show results when optimising COLD using the final state fidelity cost function C_F in the case of $N = 5$ spins for fidelity. What we find is a consistent pattern in the behaviour of the final state fidelity when using either of the two cost functions, with consistently stable final fidelities at shorter driving times and, surprisingly, better final fidelities at longer driving times, at least in the $N = 5$ case. We might attribute this to different cost function landscapes due to the different cost functions, which might have more or less optimal minima within reach for a local optimiser like Powell’s method (Sec. 3.1.3.2). Regardless, what we do find is that we can get final state fidelities consistently above 90% for a system of $N = 30$ spins while using an exponentially more efficient cost function for optimisation. While we use tensor network methods to compute the fidelities of chains with $N = 10$ spins and above in all cases, these approaches are still orders of magnitude slower at calculating a single



— DRAFT – June 28, 2023 —

Figure 7.4: Final state fidelities when control parameters are optimised via integral and amplitude cost functions of the SO LCD coefficients for the Ising spin chain with FO COLD applied to the system. In (a) we plot the results obtained when using the integral cost function $C_{I,\zeta}$ and in (c) we do the same in the case of maximum amplitude $C_{A,\zeta}$ for Ising chains of lengths $N = 2$ (pink circles), $N = 5$ (red inverted triangles) and $N = 30$ (dark red stars). We also plot results for COLD optimisation with C_F in the case of $N = 5$ spins from Fig. 6.2(a) for comparison (black crosses). We investigate how the different cost functions perform for evolution time $\tau = 0.1 J_0^{-1}$ for different system sizes N in the case where (b) FO COLD is applied to the system while minimising a SO component and (d) when applying both FO terms and one of the SO terms. For example, in (d), the pink circles are the result of implementing COLD with FO local σ^y terms and γ terms $\sigma^x \sigma^y$, $\sigma^y \sigma^x$ while minimising the maximal amplitudes of the ζ terms $\sigma^z \sigma^y$, $\sigma^y \sigma^z$. For system sizes above $N = 10$ we used ITensor [101] MPS calculations which were converged with a truncation level of 10^{-14} per time step at each site reaching a maximum bond dimension of $D = 4$. In all cases, a single optimisable parameter is used ($N_k = 1$) and the best optimisation out of 50 (lowest cost function value for each cost function) is used. Figure reproduced from [1].

iteration of the C_F cost function than in the case of either of the integral or amplitude cost functions used for the optimisation. We find that this trend is consistent, if slightly shifted depending on which AGP-based cost function is used in plot (b) of Fig. 7.4,

Chapter 7. Higher order AGP as a cost function

and that this trend is consistent when minimising one of the SO coefficients and implementing the other along with the FO terms, at least at relatively short driving times of $\tau = 0.1J_0^{-1}$.

All of these results are not conclusive proof for the advantage of using the integral or amplitude cost functions in place of C_F , especially if the LCD terms are highly local with respect to the full system size. However, the results presented do indicate a potential advantage, especially in the case of larger systems, of using knowledge about the approximate AGP operator in an optimisation procedure of the control Hamiltonian, assuming the wavefunction of the state is prohibitively difficult to access. In doing so, it is possible to sacrifice some amount of effectiveness for a large gain in efficiency and, possibly, this kind of approach could be used in cases where fidelities are simply not a tractable option in the case of numerical optimisation.

7.3 GHZ states and frustrated spins

A final example we will be considering in this chapter is GHZ state preparation scheme in a system of frustrated spins from Sec. 6.4. Thus far, both in the two-spin case and in the Ising spin chain case we have seen some evidence for advantage when it comes to using AGP-informed cost functions to optimise the control parameters with respect to the final state fidelity. While we have shown that optimising using the fidelity cost function C_F generally guarantees better results, it is also far less efficient for larger systems. Using the integral cost function C_I or the amplitude cost function C_A may not be as effective, but it is far more efficient and can be used to obtain good results for both COLD and optimal control pulses with no CD added to them. In this case, we will explore an example of a system and control pulse combination for which this option may not be viable, at least given the parameters we are working with.

We will set up the problem in the same way as in Sec. 6.4, with the bare Hamiltonian from Eq. (6.31), as well as a GRAPE control pulse as given by Eq. (6.34) and the surrounding description. In this case, from Sec. 6.4.1 we recall that we already attempted implementing a different cost function to C_F , the aim of which was to maximise a measure of tripartite entanglement in the three spin case: C_{T_3} from Eq. (6.38).

Chapter 7. Higher order AGP as a cost function

We will return to both the fidelity and the tangle as measures of success for the final state and thus we will be looking solely at the $N = 3$ spin example, both as the simplest possible example to test out new cost functions on and due to the fact that we have a non-trivial tripartite entanglement metric like the three-tangle available.

The FO LCD terms are defined as previously to be local σ^y operators scaled by the coefficient $\alpha(\beta, \mathbf{h}, \lambda)$, while the SO terms are those presented in Eq. (6.33). We will differentiate here between solving for α when only the FO terms are in the LCD ansatz, denoting this case as $\alpha^{(1)}$ and solving for all of the three coefficients α, γ and ζ as a combined FO and SO ansatz by using the coupled set of equations presented in Appendix C, wherein we will refer to the FO coefficient as $\alpha^{(2)}$ instead.

In Fig. 7.5 we plot the cost function landscapes for C_F (a-b), C_{T_3} (c-d) and several integral cost functions C_I in (e-h) for a total driving time of $\tau = 0.1J_0^{-1}$ (the results for the maximal amplitude cost functions can be found in Appendix E). The first thing that one might notice when looking at the plots is that while there appears to be some qualitative relationship between the different cost function landscapes, it is certainly not the case that there is a clear overlap between the minimum or maximum values of the integral cost functions and the highest fidelities (lowest values in the contour plot, as $C_F = 1 - F$). This is a fact that holds true in the case of the maximal amplitude cost functions in Appendix E too. The C_F and C_{T_3} landscapes are highly non-convex, with multiple local minima close to each other, something that is may be a consequence of the GRAPE cost function and the very small number of parameters, as this translates to very sudden piecewise shifts in the control pulse (see discussion in Sec. 4.2). This may also be a consequence of the degenerate nature of the ground state. It is possible, that the reason there is such a disparity between the final state fidelity and the scale of the LCD coefficients is the small number of parameters in the GRAPE function, but attempts to optimise the GRAPE control pulse with parameter numbers up to $N_k = 12$ using dual-annealing (Sec. 3.1.3.3) with C_I - and C_A -type cost functions as in Fig. 7.4 or Fig. 7.3 results in the system barely moving out of its initial state regardless of driving time (not plotted). This is certainly an indication that in a more complex regime with a degenerate ground state where the target is entanglement generation, the AGP cost

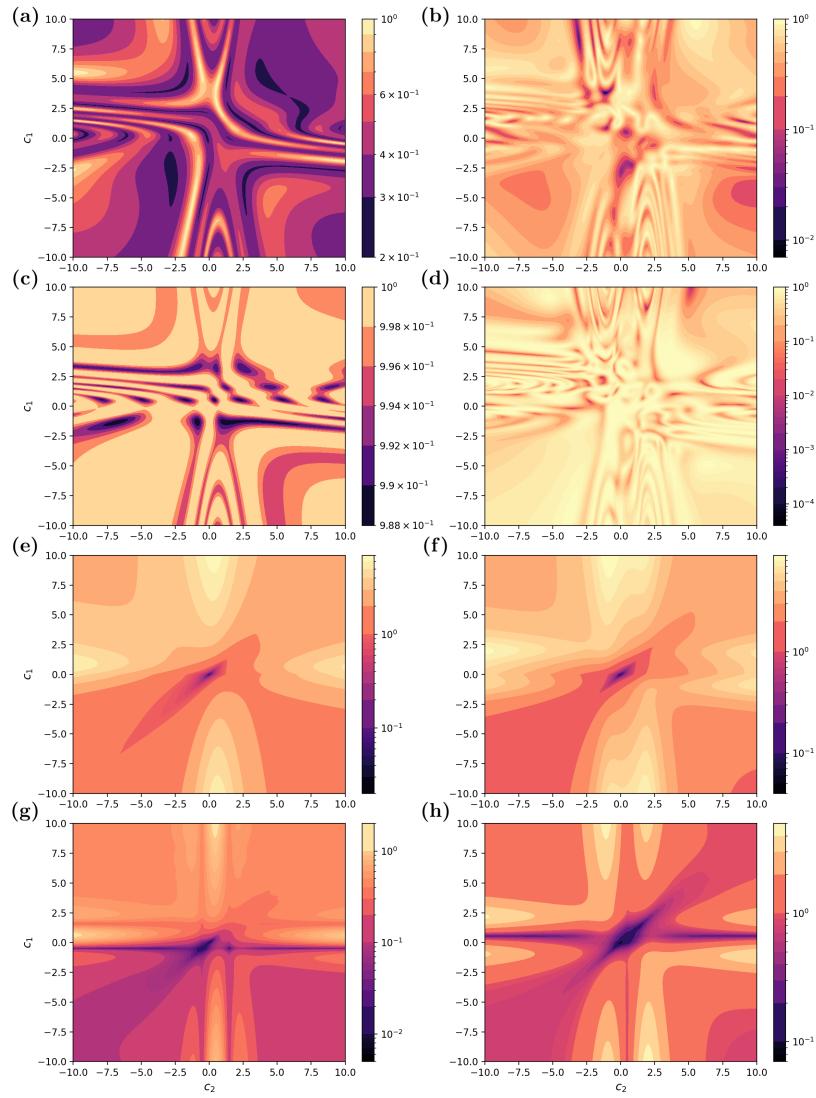


Figure 7.5: Contour plots at $\tau = 0.1J_0^{-1}$ of different cost function values for GHZ state preparation for parameters $c_1, c_2 \in [-10, 10]$ and a GRAPE control pulse. In (a) and (b) we plot C_F in the cases where FO and SO COLD is applied respectively. Then, in (c-d) we do the same for C_{T_3} , with FO COLD plotted in (c) and SO COLD plotted in (d). (e-h) are then plots of the integral cost function C_I values for the same range of parameters. In (e) we plot $C_{I,\alpha^{(1)}}$ when only FO LCD is considered, while in (f) we plot $C_{I,\alpha^{(2)}}$ as described in the text. Then in (g) we plot $C_{I,\gamma}$ and in (h) we plot $C_{I,\zeta}$, corresponding to the SO terms. Note that each plot has its own color bar, as the color encodings and the value scaling in each plot is quite different.

function approach might not work, or at least needs to be investigated further.

Reminder: mention three-spin operators!!

Part IV

Conclusion

Chapter 8

Summary

This is The End; my only friend,
The End.

Jim Morrison, “*The End*”, *The
Doors*

In this thesis we introduced a new method for speeding up adiabatic quantum protocols while minimising losses due to transitions out of the instantaneous eigenstate: “Counterdiabatic Optimised Local Driving”, or COLD. The new method is comprised of two key components: approximate counterdiabatic driving (CD) techniques and quantum optimal control. We discussed the theoretical framework and motivation behind COLD, beginning in Ch. 2 with a background introduction to quantum adiabaticity and the losses which arise as a consequence of fast changes in a time-dependent Hamiltonian. In Sec. 2.2 we covered how these losses can be described by an operator known as the adiabatic gauge potential or AGP [4], and in Sec. 2.3 we discussed how the CD technique can be used to suppress the non-adiabatic effects generated by the AGP [6,7]. We explored the reasons why the exact CD pulse is often inaccessible, either in theory or in practice, in Sec. 2.4, and introduced several existing methods for constructing an *approximate* counterdiabatic drive. Then, in Ch. 3, we covered the theory and methodology involved in optimal control theory, which concerns itself with finding optimal path for a system from some initial state to some final state. In Sec. 3.2 we introduced how optimal control theory can be applied in the setting of quantum systems and in

Chapter 8. Summary

Sec. 3.3 we introduced several popular quantum optimal control methods like CRAB and GRAPE.

These two background chapters paved the way for a detailed introduction to COLD in Part II. The new method, COLD, is the result of combining an approximate CD method we refer to as LCD [16], with quantum optimal control techniques. In Sec. 2.4.1, we outlined, how LCD works by allowing one to variationally determine a pulse shape for an ansatz set of physical operators which most closely resembles the exact counterdiabatic drive for a given time-dependent Hamiltonian. The goal of LCD is to suppress as many losses associated with such transitions as possible within the restrictions imposed by the ansatz set of operators. In Ch. 4, we described how COLD can improve upon the LCD approach by using the observation that the non-adiabatic effects experienced by a system driven by a time-dependent Hamiltonian depend on the path of the Hamiltonian through parameter space. We showed how this path can be changed via the implementation of methods from optimal control theory in Sec. 4.2, thus allowing COLD to find a path which maximises the effects of the LCD for a given ansatz set of operators. We then posited, in Ch. 5, that the information about non-adiabatic effects contained in the AGP operator could be used to construct optimisation metrics for the optimal control component of COLD.

Finally, in Part III, we demonstrated how COLD performs by numerically simulating its implementation for various physical systems and time-dependent Hamiltonians. We compared the results to those obtained when using LCD with no optimal control component, as well as to optimal control techniques with no counterdiabatic component. In Ch. 6, we focused on using optimal control techniques to target properties of the final state obtained by implementing each method, such as fidelity with respect to a target ground state or amount of entanglement. In Sec. 6.1 we showed results for a simple two-spin annealing protocol in order to demonstrate in detail how the COLD approach works. Then, in Sec. 6.2 we showed the advantage of using COLD over other approaches in the case of the Ising spin chain, even when the pulse amplitudes of all of the drives involved are constrained to be below some value, as demonstrated in Sec. 6.2.1. This was followed by a demonstration in the case of an ARP protcol

Chapter 8. Summary

for population transport in a synthetic lattice, adapted from [96] wherein only LCD had been implemented. We capped off the chapter with a more complex example, the goal of which was to generate a maximally entangled GHZ state in a system of frustrated spins. We found that COLD showed an advantage in all of these examples and that it could be enhanced with various optimal control techniques like CRAB or GRAPE. In the final example, we discovered that highly local LCD operators might not be enough to generate entanglement through the system at short driving times and that more delocalised pulses might be needed in such systems. We then demonstrated how AGP-informed cost functions, first introduced in Ch. 5, performed for some of the same systems in Ch. 7. In the case of the two-spin example and the Ising spin chain, we showed in Sec. 7.1 and Sec. 7.2 respectively that we could implement a far more efficient optimisation protocol than ones which use fidelity as a cost function for finding Hamiltonian paths that minimise non-adiabatic effects. We showed that this could be done in cases where either COLD or only optimal control is implemented. We found, however, that in the case of generating GHZ states in Sec. 7.3, such a cost function did not appear to work as intended, whether due to the complexity of the problem at hand, drawbacks of the LCD approximation or issues with the optimal control.

Chapter 9

To boldly go

Time will explain it all. He is a talker, and needs no questioning before he speaks.

Euripides

There is often joy mixed with trepidation in finding that, for all the work that might have already been done, far more remains to be accomplished. This is certainly true in the case of the results presented in this thesis. The COLD method is one that was created with practicality in mind: given a quantum system, a time-dependent Hamiltonian driving it and a set of constraints, like the system controllability or computational resources, it should help one produce an optimal protocol which minimises the non-adiabatic losses experienced by the system while it is driven from an initial eigenstate towards the target as quickly as possible. As the space of systems, Hamiltonians and constraints is vast, merely exploring in which scenarios COLD may or may not have an advantage over the equally vast set of other possible approaches is no small task. However, in this brief chapter, we will discuss several open questions and potential future research directions in a more focused way, including those that arose during the process of constructing and implementing COLD for [1].

Chapter 9. To boldly go

9.1 Physical implementation

The first thing to note is the fact that the field of quantum optimal control is very extensive and that we only explored a few common ways to construct control pulses in this thesis. In general, combining a more complex control pulse that has a larger solution space and increased computational resources will almost certainly be a better option than simpler choices, unless there is an informed reason to expect a simpler pulse to do better. In many of the examples in Ch. 6 and Ch. 7 the control pulse we implemented was the *bare* pulse from Eq. (4.5), which is quite rudimentary. One reason for doing this was to save on time and computational resources, as it required very little of either to implement compared to more complex approaches like CRAB or GRAPE. The other reason was simply the fact that the results obtained using the bare pulse were already enough to demonstrate the functionality and advantages of the method, while also being easier to analyse. In any more focused application of COLD, there would have to be a strong consideration for how a particular choice of optimal control pulse can interact with the constraints of the problem and even the LCD pulse itself, which is a function of the control parameters. A larger gradient in the control pulse could, for example, lead to a spike in the amplitude of the approximate counterdiabatic pulse, due to the $\partial_\lambda H$ matrix elements present in the AGP operator (Eq. (2.32)) or else lead to a large non-zero counterdiabatic component at the end of the protocol.

Apart from the optimal control component, it would be useful to consider the noise present in physical implementations and how that might affect the performance of COLD. The cost function landscapes plotted throughout Ch. 7 give some indication of how smoothly the final state fidelity reacts to a small shift in optimal control parameters. While in some cases, like the two-spin example of Sec. 7.1, the fidelity cost function is quite smooth, this is absolutely not the case for the GHZ state preparation example in Sec. 7.3. While the highly non-convex nature of the plots might simply be due to the small number of control parameters involved, there is no guarantee that such high susceptibility to parameter values would be avoided in any specific example.

9.2 Extensions of COLD

As well as questions following up from the existing methodology of COLD and the examples covered in this thesis, we might also look forward to new ideas inspired by the content that was presented in previous chapters. The composition of Ch. 5 was born, for example, from several observations about the behaviour of different orders of LCD operators in optimised versus un-optimised control pulses (see Appendix D for more details). In a similar vein, we can imagine designing new and better types of cost functions based on information about the non-adiabatic effects experienced by a system rather than just those presented in this thesis. The failure of the approach in the case of GHZ state preparation is certainly a reason to try something different.

A particular example of ideas inspired by COLD, we may opt to optimise the *other* component of the counterdiabatic drive: not the AGP operator, but rather the rate of change of the parameters $\dot{\lambda}$. While this may seem like a strange suggestion, as the very idea behind COLD is to increase the speed of the system evolution and thus $\dot{\lambda}$, it may be possible to perform a piecewise optimisation of how fast the changes in the Hamiltonian parameters occur at different critical moments in the system evolution. As discussed in Ch. 2, the ‘slow’ evolution condition for adiabaticity depends on the energy gaps between the instantaneous states. As such, it might be interesting to construct a control pulse which varies ∂_λ for each timestep depending on the criticality of the non-adiabatic effects experienced by the system at that point, with the full evolution being constrained to some short total evolution time τ .

Beyond these ideas and considerations, there may be better approaches to computing the approximate CD components, which is occasionally an arduous task (see e.g. Appendix C). This is the goal of some upcoming work [102], which explores how the structure in certain Hamiltonians like the Ising model can be exploited in order to compute the LCD coefficients for a large numbers of operators very efficiently. Should this be accomplished in a more general setting, AGP-based cost functions may become an even greater resource, as we might be able to better characterise the behaviour of different orders of LCD with respect to each other and the target state. The exact AGP

Chapter 9. To boldly go

operator, which not only forms the backbone of CD [4, 16], but also acts as a probe for quantum chaos [18, 103], may yet have more information for us to use in designing optimal fast adiabatic protocols.

Appendix A

Rotating spin Hamiltonian

In Chap. 2 we used the example of a spin rotating in a magnetic field to illustrate adiabatic processes in quantum systems. We considered a spin starting in the $|+\rangle$ state and being 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 $\lambda(t) = \frac{\pi t}{2\tau}$. The AGP operator ansatz \mathbb{A}_λ for this system can be described by the operator σ^y scaled by some λ -dependent coefficient which we will refer to as $\alpha(\lambda)$

$$\mathbb{A}_\lambda = \alpha(\lambda)\sigma^y \quad (\text{A.2})$$

as discussed in the main text. We will now proceed to show how we can arrive at the resulting form of α given in Eq. (2.49) using the LCD method outlined in Sec. 2.4.1.

The first step is to find the operator G_λ given in Eq. (2.43):

$$\begin{aligned} G_\lambda(\mathbb{A}_\lambda) &= \partial_\lambda H + i[\mathbb{A}_\lambda, H] \\ &= \sin(\lambda)\sigma^x - \cos(\lambda)\sigma^z + 2\alpha(\lambda)\sin(\lambda)\sigma^x - 2\alpha(\lambda)\cos(\lambda)\sigma^z \\ &= (1 + 2\alpha(\lambda))\sin(\lambda)\sigma^x - (1 + 2\alpha(\lambda))\cos(\lambda)\sigma^z, \end{aligned} \quad (\text{A.3})$$

Appendix A. Rotating spin Hamiltonian

where we have used $\hbar = 1$. This can then be used to define the action

$$\begin{aligned}\mathcal{S}(\mathbb{A}_\lambda) &= \text{Tr} [G_\lambda^2(\mathbb{A}_\lambda)] \\ &= 2(1 + 2\alpha(\lambda))^2 \sin^2(\lambda) + 2(1 + 2\alpha(\lambda))^2 \cos^2(\lambda) \\ &= 2(1 + 2\alpha(\lambda))^2.\end{aligned}\tag{A.4}$$

In order to find the form of α , we need to find the minimum of the action $\mathcal{S}(\mathbb{A}_\lambda)$ with respect to α , which can be easily done:

$$\begin{aligned}\frac{\delta \mathcal{S}}{\delta \alpha} &= 8(1 + 2\alpha(\lambda)) \\ \Rightarrow \alpha(\lambda) &= -\frac{1}{2},\end{aligned}\tag{A.5}$$

which is the expected result.

It can be checked, for this simple example, that $\mathbb{A}_\lambda = -\frac{1}{2}\sigma^y$ is the exact AGP operator \mathcal{A}_λ by using Eq. (2.32) where the matrix elements of the AGP are written out explicitly as:

$$\begin{aligned}\mathcal{A}_\lambda &= i \left(\sum_n \langle n | \partial_\lambda n \rangle |n\rangle\langle n| + \sum_{m \neq n} |m\rangle \frac{\langle m | \partial_\lambda H | n \rangle}{(E_n - E_m)} \langle n | \right) \\ &= i \left(\sum_n \langle n | \partial_\lambda n \rangle |n\rangle\langle n| + \sum_{m \neq n} \langle m | \partial_\lambda n \rangle |m\rangle\langle n| \right)\end{aligned}\tag{A.6}$$

In this case, the adiabatic eigenstates of the Hamiltonian $H(\lambda)$ are

$$\begin{aligned}|\psi_1\rangle &= \frac{1}{n_1} [(\sec(\lambda) + \tan(\lambda)) |\uparrow\rangle + |\downarrow\rangle] \\ |\psi_2\rangle &= \frac{1}{n_2} [(-\sec(\lambda) + \tan(\lambda)) |\uparrow\rangle + |\downarrow\rangle],\end{aligned}\tag{A.7}$$

where $n_1 = \sqrt{1 + |\sec(\lambda) + \tan(\lambda)|^2}$ and $n_2 = \sqrt{1 + |-\sec(\lambda) + \tan(\lambda)|^2}$ are the normalisation factors. Their derivatives with respect to λ are

$$\begin{aligned}|\partial_\lambda \psi_1\rangle &= \frac{1}{n_1^3} \sec(\lambda) [(\sec(\lambda) + \tan(\lambda)) |\uparrow\rangle - (\sec(\lambda) + \tan(\lambda))^2 |\downarrow\rangle] \\ |\partial_\lambda \psi_2\rangle &= (2 + 2 \sin(\lambda))^{-3/2} [(\sin(\lambda) + 1)^2 |\uparrow\rangle + \cos(\lambda)(1 + \sin(\lambda)) |\downarrow\rangle].\end{aligned}\tag{A.8}$$

Appendix A. Rotating spin Hamiltonian

Evaluating $\langle \psi_1 | \partial_\lambda \psi_1 \rangle$ and $\langle \psi_2 | \partial_\lambda \psi_2 \rangle$ we find that they are equal to 0, meaning the diagonal elements of \mathcal{A}_λ are 0. Doing the same for the off-diagonals we find

$$\begin{aligned}\langle \psi_1 | \partial_\lambda \psi_2 \rangle &= \frac{1}{2} \\ \langle \psi_2 | \partial_\lambda \psi_1 \rangle &= -\frac{1}{2},\end{aligned}\tag{A.9}$$

meaning that the exact AGP operator, as found from evaluating its matrix elements is just

$$\mathcal{A}_\lambda = \begin{pmatrix} 0 & \frac{i}{2} \\ -\frac{i}{2} & 0 \end{pmatrix} = \alpha(\lambda) \sigma^y,\tag{A.10}$$

which is the result obtained previously from the LCD approach.

Appendix B

Pontryagin maximum principle

Theorem 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*

$$\dot{\lambda}^T = -\lambda^T f(x(t), u(t)) \quad (\text{B.1})$$

with terminal condition

$$\lambda^T(\tau) = -\phi(x(\tau)) \quad (\text{B.2})$$

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

$$\lambda^T(t)f(x(t), u(t)) \geq \lambda^T(t)f(x(t), v) \quad (\text{B.3})$$

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

$$\lambda^T(t)f(x(t), u(t)) = c, \quad (\text{B.4})$$

for a constant c

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

$$h(\lambda, x, u) := \lambda^T(t)f(x, u). \quad (\text{B.5})$$

Appendix B. Pontryagin maximum principle

Now we can recast Eqs. (B.3) and (B.4):

$$\begin{aligned} h(\lambda(t), x(t), u(t)) &= c \\ h(\lambda, x, u) &\geq h(\lambda, x, v), \end{aligned} \tag{B.6}$$

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

$$\begin{aligned} \dot{x} &= f(x, u(x, \lambda)), \\ \dot{\lambda}^T &= -\lambda^T f(x, u(x, \lambda)) \end{aligned} \tag{B.7}$$

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.

Appendix C

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:

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

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. 6.4.

The first order LCD ansatz, as stated in the main text, is just single-spin operators:

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

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

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

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

The first order commutators are computed as follows:

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

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

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

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 C. Derivation of the CD coefficients for an arbitrary Ising graph

per Eq. (2.43)):

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

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 C. Derivation of the CD coefficients for an arbitrary Ising graph

to each α_i gives:

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

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

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

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

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

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

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

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

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

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

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 D

Additional plots for the Ising spin chain example

This appendix contains additional information and plots concerning the Ising spin chain example from Sec. 6.2, where we investigate implementations of COLD, LCD and BPO for this particular system. In the plots presented here, all parameters are the same as those discussed in Sec. 6.2 unless stated otherwise.

As well as the five spin chain which is analysed in detail in the, in Fig. D.1 we also present a plot of how FO COLD and BPO scale with (a) increasing chain lengths and (b) increasing number of control parameters N_k when the bare optimisation pulse from Eq. (6.28) is used on $N = 5$ spins.

Appendix D. Additional plots for the Ising spin chain example

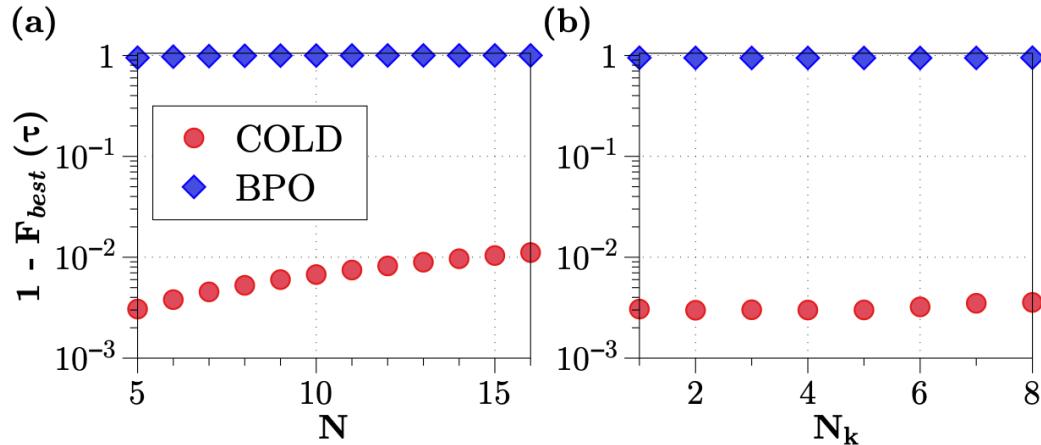


Figure D.1: Scaling of fidelities in the annealing protocol for the Ising model with (a) system size N and (b) optimisation parameters N_k at driving time $\tau = 10^{-2} J_0^{-1}$. Plots show a comparison between BPO (blue diamonds) and COLD (red circles). In (a) we see that the COLD fidelity decreases as a function of N but remains quite high when compared to BPO while (b) shows the non-existent improvement for both BPO and COLD with an increasing number of parameters in the $N = 5$ spin case. Plotted best fidelities are obtained across 500 optimisations. Figure reproduced from [1].

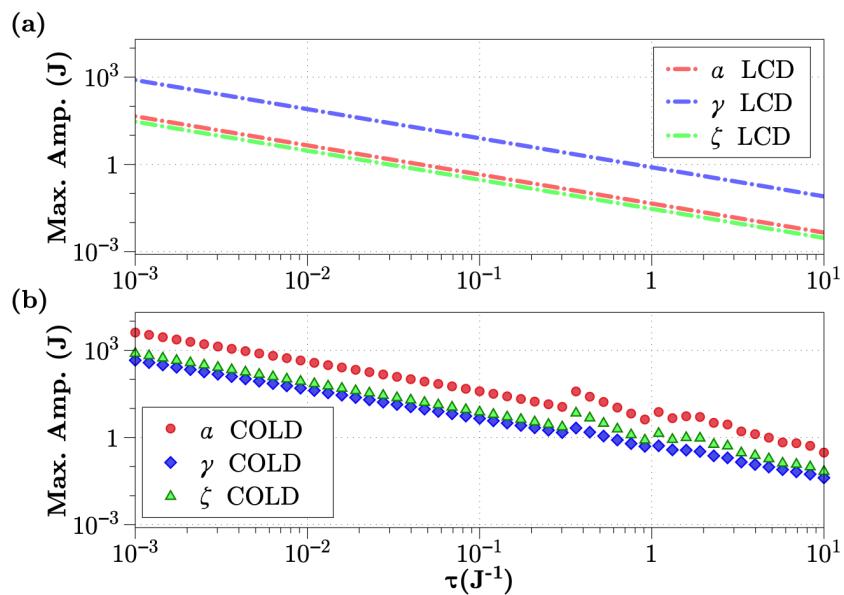


Figure D.2: Figure reproduced from [1].

Appendix E

More details and plots for the higher order AGP chapter

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

— DRAFT – June 28, 2023 —

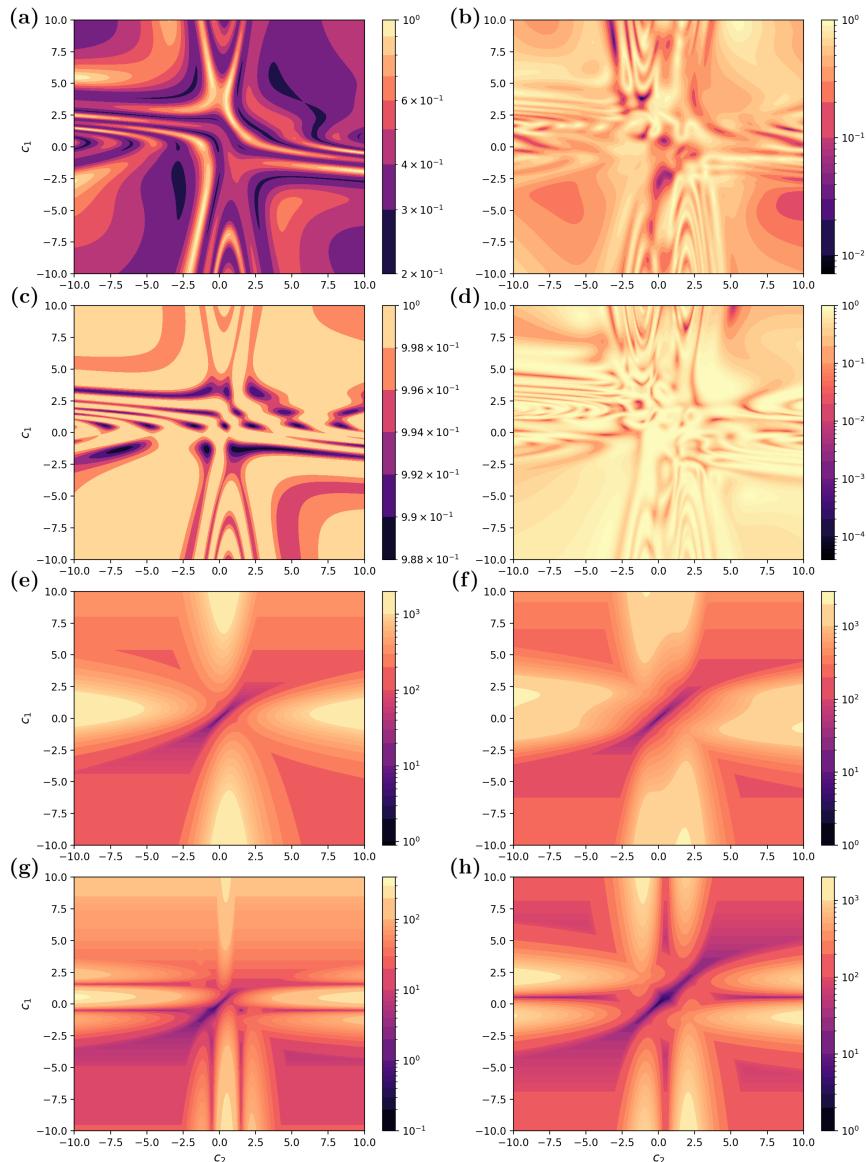


Figure E.1: Contour plots at $\tau = 0.1J_0^{-1}$ of different cost function values for GHZ state preparation for parameters $c_1, c_2 \in [-10, 10]$ and a GRAPE control pulse. In (a) and (b) we plot C_F in the cases where FO and SO COLD is applied respectively. Then, in (c-d) we do the same for C_{T_3} , with FO COLD plotted in (c) and SO COLD plotted in (d). (e-h) are then plots of the integral cost function C_I values for the same range of parameters. In (e) we plot $C_{I,\alpha^{(1)}}$ when only FO LCD is considered, while in (f) we plot $C_{I,\alpha^{(2)}}$ as described in the text. Then in (g) we plot $C_{I,\gamma}$ and in (h) we plot $C_{I,\zeta}$, corresponding to the SO terms. Note that each plot has its own color bar, as the color encodings and the value scaling in each plot is quite different.

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

— DRAFT – June 28, 2023 —

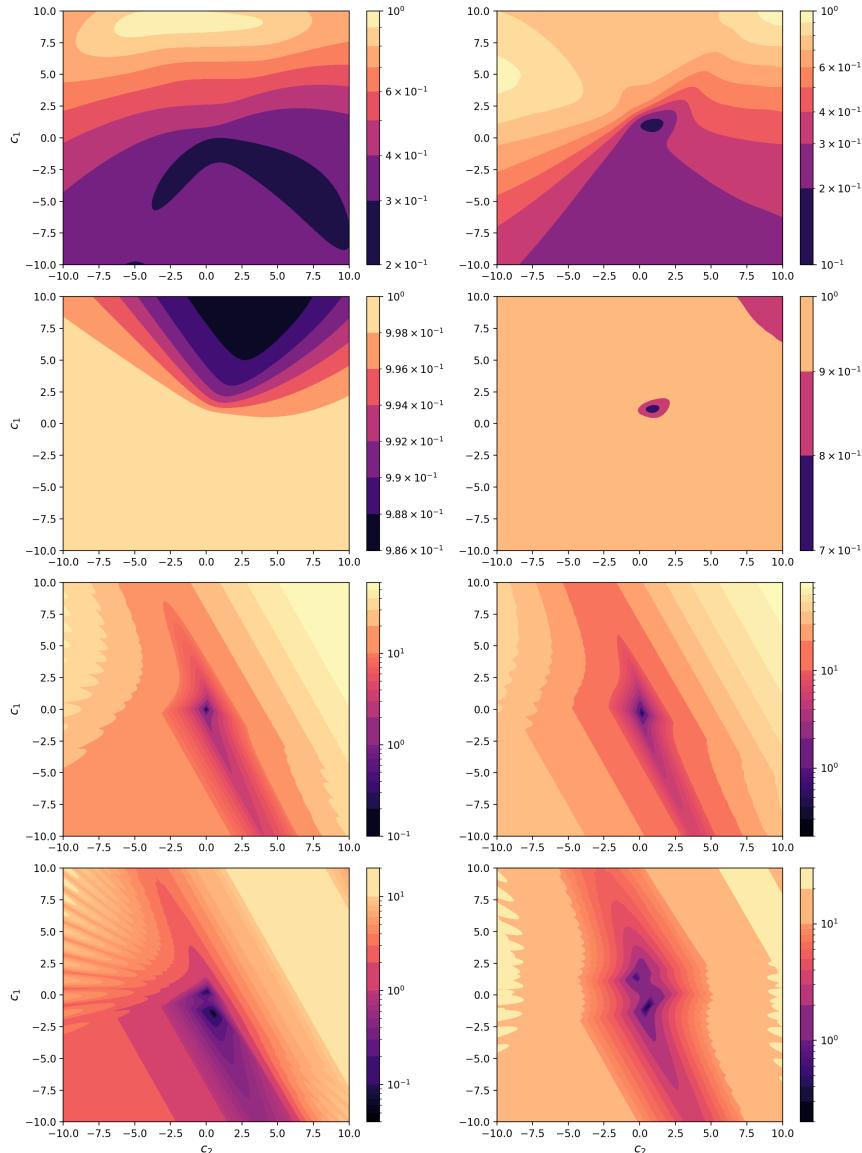


Figure E.2: Contour plots at $\tau = 0.1J_0^{-1}$ of different cost function values for GHZ state preparation for parameters $c_1, c_2 \in [-10, 10]$ and a GRAPE control pulse. In (a) and (b) we plot C_F in the cases where FO and SO COLD is applied respectively. Then, in (c-d) we do the same for C_{T_3} , with FO COLD plotted in (c) and SO COLD plotted in (d). (e-h) are then plots of the integral cost function C_I values for the same range of parameters. In (e) we plot $C_{I,\alpha^{(1)}}$ when only FO LCD is considered, while in (f) we plot $C_{I,\alpha^{(2)}}$ as described in the text. Then in (g) we plot $C_{I,\gamma}$ and in (h) we plot $C_{I,\zeta}$, corresponding to the SO terms. Note that each plot has its own color bar, as the color encodings and the value scaling in each plot is quite different.

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

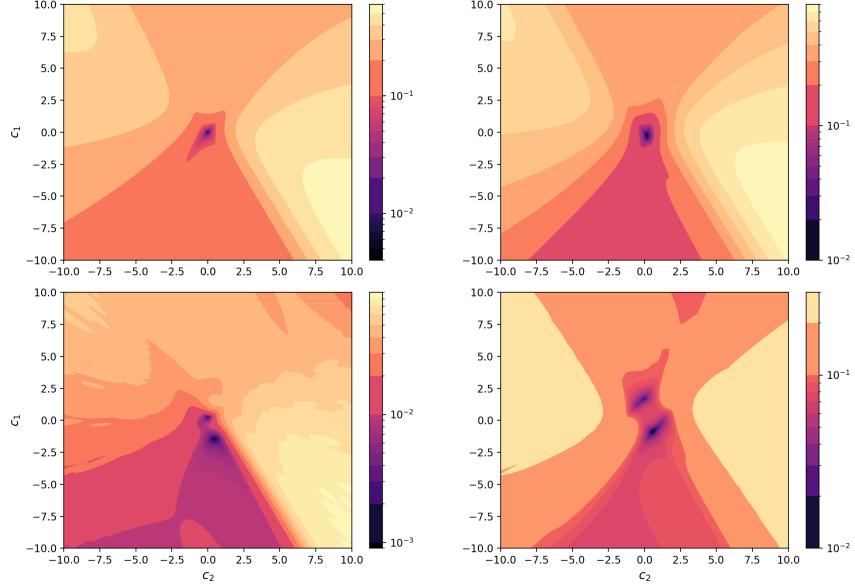


Figure E.3: Contour plots at $\tau = 0.1J_0^{-1}$ of different cost function values for GHZ state preparation for parameters $c_1, c_2 \in [-10, 10]$ and a GRAPE control pulse. In (a) and (b) we plot C_F in the cases where FO and SO COLD is applied respectively. Then, in (c-d) we do the same for C_{T_3} , with FO COLD plotted in (c) and SO COLD plotted in (d). (e-h) are then plots of the integral cost function C_I values for the same range of parameters. In (e) we plot $C_{I,\alpha^{(1)}}$ when only FO LCD is considered, while in (f) we plot $C_{I,\alpha^{(2)}}$ as described in the text. Then in (g) we plot $C_{I,\gamma}$ and in (h) we plot $C_{I,\zeta}$, corresponding to the SO terms. Note that each plot has its own color bar, as the color encodings and the value scaling in each plot is quite different.

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,

Bibliography

- no. 24, p. 245147, Dec. 2022, arXiv:2206.08599 [cond-mat, physics:hep-th, physics:quant-ph]. [Online]. Available: <http://arxiv.org/abs/2206.08599>
- [20] F. Petiziol, “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. Cykert, 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,

Bibliography

- no. 1, pp. 1–60, Dec. 2022, number: 1 Publisher: SpringerOpen. [Online]. Available: <https://epjquantumtechnology.springeropen.com/articles/10.1140/epjqt/s40507-022-00138-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

Bibliography

- and 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] J. Nocedal and S. Wright, *Numerical Optimization*. Springer Science & Business Media, Dec. 2006, google-Books-ID: VbHYoSyelFcC.
- [44] 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>
- [45] L. Bottou and O. Bousquet, “The Tradeoffs of Large Scale Learning,” in *Advances in Neural Information Processing Systems*, vol. 20. Curran Associates,

Bibliography

- Inc., 2007. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2007/hash/0d3180d672e08b4c5312dcdafdf6ef36-Abstract.html
- [46] 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
- [47] 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.
- [48] 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>
- [49] 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>
- [50] 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>
- [51] R. P. Brent, *Algorithms for Minimization Without Derivatives*. Courier Corporation, Jan. 2002.
- [52] 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>

Bibliography

- [53] 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>
- [54] 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>
- [55] 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>
- [56] G. C. Hegerfeldt, “Driving at the Quantum Speed Limit: Optimal Control of a Two-Level System,” *Physical Review Letters*, vol. 111, no. 26, p. 260501, Dec. 2013, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.111.260501>
- [57] P. M. Poggi, F. C. Lombardo, and D. A. Wisniacki, “Quantum speed limit and optimal evolution time in a two-level system,” *Europhysics Letters*, vol. 104, no. 4, p. 40005, Dec. 2013, publisher: EDP Sciences, IOP Publishing and Società Italiana di Fisica. [Online]. Available: <https://dx.doi.org/10.1209/0295-5075/104/40005>
- [58] 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>
- [59] 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>

Bibliography

- [60] 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>
- [61] 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>
- [62] ——, “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>
- [63] 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>
- [64] 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>
- [65] 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>
- [66] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, “Optimal control of coupled spin dynamics: design of NMR

Bibliography

- 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>
- [67] 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>
- [68] 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>
- [69] 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>
- [70] 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>
- [71] F. Motzoi, J. M. Gambetta, S. T. Merkel, and F. K. Wilhelm, “Optimal control methods for rapidly time-varying Hamiltonians,” *Physical Review A*, vol. 84, no. 2, p. 022307, Aug. 2011, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.84.022307>
- [72] B. Anderson, H. Sosa-Martinez, C. Riofrío, I. H. Deutsch, and P. S. Jessen, “Accurate and Robust Unitary Transformations of a High-Dimensional Quantum System,” *Physical Review Letters*, vol. 114, no. 24, p. 240401,

Bibliography

- DRAFT - June 28, 2023 -

Jun. 2015, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.114.240401>

- [73] 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>
- [74] O. V. Morzhin and A. N. Pechen, “Krotov method for optimal control of closed quantum systems,” *Russian Mathematical Surveys*, vol. 74, no. 5, p. 851, Oct. 2019, publisher: IOP Publishing. [Online]. Available: <https://iopscience.iop.org/article/10.1070/RM9835/meta>
- [75] R. Chakrabarti and H. Rabitz, “Quantum control landscapes,” *International Reviews in Physical Chemistry*, vol. 26, no. 4, pp. 671–735, Oct. 2007, publisher: Taylor & Francis. [Online]. Available: <https://www.tandfonline.com/doi/citedby/10.1080/01442350701633300>
- [76] H. Rabitz, B. Russell, and T.-S. Ho, “The Surprising Ease of Finding Optimal Solutions for Controlling Nonlinear Phenomena in Quantum and Classical Complex Systems,” *The Journal of Physical Chemistry A*, vol. 127, no. 19, pp. 4224–4236, May 2023, publisher: American Chemical Society. [Online]. Available: <https://doi.org/10.1021/acs.jpca.3c01896>
- [77] D. Dong and I. R. Petersen, “Quantum estimation, control and learning: Opportunities and challenges,” *Annual Reviews in Control*, vol. 54, pp. 243–251, Jan. 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1367578822000281>
- [78] A. d. Campo, J. Goold, and M. Paternostro, “More bang for your buck: Super-adiabatic quantum engines,” *Scientific Reports*, vol. 4, p. 6208, Aug. 2014. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4147366/>
- [79] H. Saberi, T. Opatrný, K. Mølmer, and A. del Campo, “Adiabatic tracking of quantum many-body dynamics,” *Physical Review A*, vol. 90, no. 6, p.

Bibliography

- 060301, Dec. 2014, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.90.060301>
- [80] S. Campbell, G. De Chiara, M. Paternostro, G. M. Palma, and R. Fazio, “Shortcut to Adiabaticity in the Lipkin-Meshkov-Glick Model,” *Physical Review Letters*, vol. 114, no. 17, p. 177206, May 2015, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.114.177206>
- [81] C. Whitty, A. Kiely, and A. Ruschhaupt, “Quantum control via enhanced shortcuts to adiabaticity,” *Physical Review Research*, vol. 2, no. 2, p. 023360, Jun. 2020. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevResear ch.2.023360>
- [82] M. Bukov, D. Sels, and A. Polkovnikov, “Geometric Speed Limit of Accessible Many-Body State Preparation,” *Physical Review X*, vol. 9, no. 1, p. 011034, Feb. 2019. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevX.9.011034>
- [83] S. Geier, N. Thaicharoen, C. Hainaut, T. Franz, A. Salzinger, A. Tebben, D. Grimshndl, G. Zürn, and M. Weidmüller, “Floquet Hamiltonian engineering of an isolated many-body spin system,” *Science*, vol. 374, no. 6571, pp. 1149–1152, Nov. 2021, publisher: American Association for the Advancement of Science. [Online]. Available: <https://www.science.org/doi/10.1126/science.abd9547>
- [84] “Spline interpolation - Encyclopedia of Mathematics.” [Online]. Available: https://encyclopediaofmath.org/index.php?title=Spline_interpolation
- [85] T. Hatomura and K. Takahashi, “Controlling and exploring quantum systems by algebraic expression of adiabatic gauge potential,” *Physical Review A*, vol. 103, no. 1, p. 012220, Jan. 2021. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.103.012220>
- [86] T. Albash and D. A. Lidar, “Adiabatic quantum computation,” *Reviews of Modern Physics*, vol. 90, no. 1, p. 015002, Jan. 2018, publisher: American

Bibliography

- Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/RevModPhys.90.015002>
- [87] I. R. Petersen and B. Ross Barmish, “Control effort considerations in the stabilization of uncertain dynamical systems,” *Systems & Control Letters*, vol. 9, no. 5, pp. 417–422, Nov. 1987. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0167691187900715>
- [88] N. Mohseni, P. L. McMahon, and T. Byrnes, “Ising machines as hardware solvers of combinatorial optimization problems,” Apr. 2022, arXiv:2204.00276 [physics, physics:quant-ph]. [Online]. Available: <http://arxiv.org/abs/2204.00276>
- [89] N. Lang and H. P. Büchler, “Topological networks for quantum communication between distant qubits,” *npj Quantum Information*, vol. 3, no. 1, pp. 1–10, Nov. 2017, number: 1 Publisher: Nature Publishing Group. [Online]. Available: <https://www.nature.com/articles/s41534-017-0047-x>
- [90] Y. Hatsugai, “Chern number and edge states in the integer quantum Hall effect,” *Physical Review Letters*, vol. 71, no. 22, pp. 3697–3700, Nov. 1993, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.71.3697>
- [91] D. Hügel and B. Paredes, “Chiral ladders and the edges of quantum Hall insulators,” *Physical Review A*, vol. 89, no. 2, p. 023619, Feb. 2014, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.89.023619>
- [92] R. Landig, L. Hruby, N. Dogra, M. Landini, R. Mottl, T. Donner, and T. Esslinger, “Quantum phases from competing short- and long-range interactions in an optical lattice,” *Nature*, vol. 532, no. 7600, pp. 476–479, Apr. 2016, number: 7600 Publisher: Nature Publishing Group. [Online]. Available: <https://www.nature.com/articles/nature17409>
- [93] T. Keller, S. B. Jäger, and G. Morigi, “Phases of cold atoms interacting via photon-mediated long-range forces,” *Journal of Statistical Mechanics: Theory*

Bibliography

- and Experiment*, vol. 2017, no. 6, p. 064002, Jun. 2017, publisher: IOP Publishing and SISSA. [Online]. Available: <https://dx.doi.org/10.1088/1742-5468/aa71d7>
- [94] M. Baranov, L. Dobrek, K. Goral, L. Santos, and M. Lewenstein, “Ultracold Dipolar Gases – a Challenge for Experiments and Theory,” *Physica Scripta*, vol. 2002, no. T102, p. 74, Jan. 2002, publisher: IOP Publishing. [Online]. Available: <https://iopscience.iop.org/article/10.1238/Physica.Topical.102a00074/meta>
- [95] C. Trefzger, C. Menotti, B. Capogrosso-Sansone, and M. Lewenstein, “Ultracold dipolar gases in optical lattices,” *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 44, no. 19, p. 193001, Sep. 2011. [Online]. Available: <https://dx.doi.org/10.1088/0953-4075/44/19/193001>
- [96] E. J. Meier, K. Ngan, D. Sels, and B. Gadway, “Counterdiabatic control of transport in a synthetic tight-binding lattice,” *Physical Review Research*, vol. 2, no. 4, p. 043201, Nov. 2020, arXiv: 2005.07052. [Online]. Available: <http://arxiv.org/abs/2005.07052>
- [97] D. M. Greenberger, M. A. Horne, A. Shimony, and A. Zeilinger, “Bell’s theorem without inequalities,” *American Journal of Physics*, vol. 58, no. 12, pp. 1131–1143, Dec. 1990. [Online]. Available: <https://doi.org/10.1119/1.16243>
- [98] 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>
- [99] A. Cabello, “Bell’s theorem with and without inequalities for the three-qubit Greenberger-Horne-Zeilinger and W states,” *Physical Review A*, vol. 65, no. 3, p. 032108, Feb. 2002, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.65.032108>
- [100] V. Coffman, J. Kundu, and W. K. Wootters, “Distributed entanglement,” *Physical Review A*, vol. 61, no. 5, p. 052306, Apr. 2000, publisher: American

Bibliography

Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.61.052306>

- [101] M. Fishman, S. White, and E. Stoudenmire, “The ITensor Software Library for Tensor Network Calculations,” *SciPost Physics Codebases*, p. 4, Aug. 2022. [Online]. Available: <https://scipost.org/10.21468/SciPostPhysCodeb.4>
- [102] E. D. Lawrence, S. Schmid, I. Čepaitė, P. Kriton, and C. W. Duncan, “Numerical approaches for non-adiabatic terms in quantum annealing,” 2023.
- [103] B. Bhattacharjee, “A Lanczos approach to the Adiabatic Gauge Potential,” Feb. 2023, arXiv:2302.07228 [cond-mat, physics:hep-th, physics:quant-ph]. [Online]. Available: <http://arxiv.org/abs/2302.07228>

Bibliography

Bibliography