Numerical quantum mechanics and A Gentle Introduction to Quantum Optimal Control Theory

Scope
This note aims to first introduce the main ideas behind numerical simulations of quantum mechanics, including the analytical considerations that makes it feasible. Specifically the note examines the physics of a single particle in 1D. Subsequently we formulate the state transfer problem of Quantum Optimal Control Theory (QOCT). Specifically we treat the *Transport* problem in great detail as an example.

Remarks

If you find any typos, inconsistencies or errors, please let me know.

Functions like $\psi = \psi(x,t)$ will denote their arguments only (sometimes just 1) when important in the context. This is an attempt to avoid clutter. While this is ultimately a subjective opinion, the reader can use it to infer what the author thinks is relevant.

The note will be mathematical only to the extent that it makes sense from a physical perspective, i.e. no math for the sake of math.

The notation will also focus on conveying ideas instead of being strictly mathematically rigorous.

Writing numerals such as 1,2,3 over "one", "two", "three" is purposefully prefered.

Units: $\hbar = m = 1$ throughout the note (chapter 5).

Notation
$\partial_t \psi = \frac{\partial \psi}{\partial_t} = \dot{\psi}$
$[a:b:c] = [a, a+b, a+2b, \cdots a+Nb]$ where $a+Nb = c$

Vector notation $\vec{v}$

Matrix notation $\mathbf{M}$

Real part of complex number $\mathcal{R}(\kappa) = \mathcal{R}(a+ib) = a$

Braket inner products $\langle\phi|\psi\rangle = \int_{-\infty}^{+\infty} \phi(x)^* \psi(x) dx$

1 Introduction

You have learned [1] how to solve the dynamics of a quantum mechanical problem in an analytical pen and paper style fashion. For 1 particle in 1D, the problems you have seen has been specified by some potential $V(x)$. The dynamics, or time evolution, of such a system is governed by the ubiquitous time dependent Schrödinger equation

$$i\dot{\psi} = \hat{H}\psi = \left[-\frac{1}{2}\frac{\partial^2}{\partial x^2} + V(x)\right]\psi \tag{1.1}$$

To solve this partial differential equation, the strategy has been to first solve the easier time independent Schrödinger equation

$$\hat{H}\phi_n = E_n\phi_n \tag{1.2}$$

where $\{\phi_n, E_n\}$ is the n'th eigenstate-eigenvalue pair belonging to $\hat{H}$. The wave function can be expanded on these eigenstates

$$\psi(x,t) = \sum_{n=0}^{\infty} c_n(t)\phi_n(x) \tag{1.3}$$

where the time dependence lies *only* in the coefficients $c_n(t)$. Inserting eq. 1.3 into eq. 1.1 and using Fourier's trick results in the (almost simplest possible) differential equation for $c_n(t)$

$$\dot{c}_n(t) = -iE_nc_n(t) \Rightarrow c_n(t) = c_n(0)e^{-iE_nt} \tag{1.4}$$

Having obtained the coefficients $c_n(t)$ we simply insert the result in eq. 1.3 to obtain a closed analytical form of the wave function for any time t

$$\psi(x,t) = \sum_{n=0}^{\infty} c_n(0)\phi_n(x)e^{-iE_nt} \tag{1.5}$$

The only thing left is to supply the inital coefficients $c_n(0)$ to match the initial state $\psi(x,0)$

$$c_n(0) = \langle\phi_n|\psi(0)\rangle = \int_{-\infty}^{+\infty} \phi_n^*(x)\psi(x,0)dx \tag{1.6}$$

From a general standpoint we are done – great!

In 1 page and 5 equations we have developed a cookbook recipe that solves all dynamical quantum mechanics, so why even bother with a numerical approach?

First of all, because the statement is obviously fairly ridiculous, for at least 2 reasons:

1. We have to be able to analytically solve the time independent Schrödinger equation for $\{\phi_n, E_n\}$
2. We have restricted ourselves to only solving problems of the form $V(x)$, i.e. static potentials with no time dependence.

To make matters worse, the two are not mutually exclusive. If I gave you a potential

$$V(x,t) = \begin{cases} ax + 0.5\cos(bx)\cdot(x-\sin(\omega t))^2 - Ae^{-\frac{(x-x_0)^2}{2(\sigma+\sin(\omega t))^2}} & \text{if } -L \le x \le +L \\ \infty & \text{otherwise} \end{cases} \tag{1.7}$$

and ordered you to give me a closed analytical expression for $\psi(x,t)$ you would probably just pack your things and go home. This example is, of course, a bit contrived, but the take-home message remains valid.

When you followed the recipe, you did it for very simplistic and idealized potentials, such as the infinite square well, and the harmonic oscillator. Alas, these potentials are only a drop in the ocean of all possible potentials. The numerical approach, on the other hand, accepts any $V(x,t)$ whatsoever. We can solve all dynamical quantum mechanics, so why even bother with an analytical approach?

It only took 0.5 page, 1 equation, and a few arguments to flip the question. So what side of the fence are we on? The answer is both. While the shortcomings of analytical methods are indeed existent, they were also deliberatly pitted against some of the strongest characteristics of the numerical approach[1]. Approximation schemes such as pertubation theory, and the WKB approximation expands the scope of what can be done with analytical tools. In addition, calling something a "numerical approach" can also be misleading, because more often than not, these methods also employ analytical insight in some shape or form. In fact, if we used no analytical knowledge whatsoever, the objectives we seek to solve in this note would be infeasible. The way we end up describing time evolution is also very useful as an analytical tool that is extremely handy.

Conclusively, analytical and numerical approaches should be thought of as complementary, and we will think of the latter as the case where we have used up every bit of analytical knowledge, and now we need the computer to fill in the numbers.

The remainder of the note is structured as follows:
Chapter 2: We will cover the basic numerical methods that allows us to simulate time evolution for wave functions in arbitrarily complex potentials, such as the one in eq. 1.7. Interestingly we do not actually need to calculate $\{\phi_n, E_n\}$ to do this!
Chapter 3: We will introduce components and develop the language of QOCT by looking at the simple *Transport* problem, in which we attempt to steer an atom initially located at position A to position B by using some *control* function.

Chapter 4: We will motivate and derive the simplest QOCT algorithm, GRAPE, which allows us to iteratively improve our control function, such that each iteration brings us closer to our objective described in Ch. 3. Finally we will discuss how to improve our the GRAPE algorithm substantially, without getting too technical.

Chapter 5: In this stand-alone chapter I go into detail with explaining what a statement like "$\hbar = m = 1$" actually means and how it works. I discuss units and the process of non-dimensionalization.

2 Representing Quantum Mechanics on a Computer

2.1 Discretization

The wave function $\psi(x,t)$ for a particle in 1D is a continuous, complex function that accepts any pair of coordinates in space and time ($x \in \mathbb{R}, t \in \mathbb{R}$). The probability for the particle to be located in the interval $[x, x+dx]$ at time t is

$$|\psi(x,t)|^2 dx \tag{2.1}$$

and we have the usual normalization condition

$$\int_{-\infty}^{\infty} |\psi(x,t)|^2 dx = 1 \tag{2.2}$$

for all t.

There are infinitely many x's and t's, and we cannot store infinity on a computer, so instead we make the intuitive approximation of discretizing both space and time on a grid:

$$x \in [x_{min} : \Delta x : x_{max}] \tag{2.3}$$

$$t \in [t_0 : \Delta t : T] \tag{2.4}$$

[1]To justify the existence of this note

It is useful to arrange the discrete values in vectors $\vec{x}$ and $\vec{t}$.
We will denote number of elements in $\vec{x}$ and $\vec{t}$ as

$$N = \text{length}(\vec{x}) \tag{2.5}$$

$$M = \text{length}(\vec{t}) \tag{2.6}$$

The main consequence of the x-discretization is that

a) Wave functions[2] turn into vectors $\in \mathbb{C}^N$

b) Operators turn into matrices $\in \mathbb{C}^{N\times N}$

The main consequence of the t-discretization becomes clear when we discuss time evolution.

2.1.1 Discretization – Wave functions

Under the x- and t-discretization approximation the wave function also becomes discrete in space and time, and we may arrange it in a vector as

The Discrete Wave Function

$$\vec{\psi}(t_j) = \begin{pmatrix} \psi_1(t_j) \\ \vdots \\ \psi_i(t_j) \\ \vdots \\ \psi_N(t_j) \end{pmatrix} \in \mathbb{C}^N, \qquad \begin{matrix} i \in [1:1:\text{length}(\vec{x})] \\ j \in [1:1:\text{length}(\vec{t})] \end{matrix} \tag{2.7}$$

Starting from some initial wave function at $\vec{\psi} = \vec{\psi}(t_1)$ we calculate the update

$$\vec{\psi}(t_1) \quad \xrightarrow{\Delta t} \quad \vec{\psi}(t_2) \quad \xrightarrow{\Delta t} \quad \dots \quad \xrightarrow{\Delta t} \quad \vec{\psi}(T) \tag{2.8}$$

by solving the time dependent Schrödinger equation at each time step. We will defer the exact details of how this is done until we know how to describe the Hamiltonian in the discretized language. This notation is very useful, because we often only care about either the instantaneous wave function at $t = t_j$ or at $t = T$. As a code implementation detail we could repeatedly overwrite the $\vec{\psi}$ vector with the update for each time step. If we did want the entire history of $\vec{\psi}$ we just store it in a matrix

$$\psi = \begin{bmatrix} \vec{\psi}(t_1) & \dots & \vec{\psi}(T) \end{bmatrix} \tag{2.9}$$

Under this discretization the probability that the particle is located within the interval $[x_i, x_i + \Delta x]$ at $t = t_j$ is

$$|\psi_i(t_j)|^2 \Delta x \tag{2.10}$$

and the discrete version of the normalization condition

$$\sum_{i=1}^{N} |\psi_i(t_j)|^2 \Delta x = 1 \tag{2.11}$$

2.1.2 Discretization – Operators

We now turn our attention to the effect of discretization of operators. Earlier it was claimed that operators become matrices after discretization. We will now argue why.

In general, any operator A acting on some wave function ψ produces a new wave function ϕ

$$\phi = A\psi \tag{2.12}$$

where ϕ and ψ are elements of the same vector space, and A is an operator on the vector space:

[2]The wave functions were already vectors in a linear algebra sense, albeit in another vector space

- If ψ is a continuous complex function, then ϕ is also a continuous complex function
- If ψ is discretized as a complex vector $\in \mathbb{C}^{\mathbb{N}}$, then $\phi \in \mathbb{C}^{\mathbb{N}}$

For the second statement to be true, A must be either a scalar or a $N \times N$ matrix. We will see that the latter is the case.

In particular we are interested in discretizing the Hamiltonian operator

$$\hat{H} = \hat{T} + \hat{V} = -\frac{1}{2}\frac{\partial^2}{\partial x^2} + V(x) \tag{2.13}$$

where V depends only on x[3]. We treat the 2 terms in turn, starting with V, and then add them together.

V: If we examine equation 2.12 with $\hat{A} = V(x)$ then the i'th entry in the resulting vector $\vec{\phi}$ is

$$\phi_i = V_i\psi_i, \qquad i \in [1:1:N] \tag{2.14}$$

where $V_i = V(\vec{x}_i)$. The entire vector $\vec{\phi}$ is then

The Potential Energy Matrix

$$\vec{\phi} = \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_i \\ \vdots \\ \phi_N \end{pmatrix} = \begin{pmatrix} V_1\psi_1 \\ \vdots \\ V_i\psi_i \\ \vdots \\ V_N\psi_N \end{pmatrix} = \begin{bmatrix} V_1 & 0 & \dots & \dots & 0 \\ 0 & \ddots & & & \vdots \\ \vdots & & V_i & & \vdots \\ \vdots & & & \ddots & 0 \\ 0 & \dots & \dots & 0 & V_N \end{bmatrix} \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_i \\ \vdots \\ \psi_N \end{pmatrix} \equiv \mathbf{V}\vec{\psi} \tag{2.15}$$

The potential is a diagonal $N \times N$ matrix $\mathbf{V}$ with the i'th diagonal element being $V_i = V(\vec{x}_i)$.

T: We now examine equation 2.12 with $\hat{A} = T$, which requires us to find a discrete version of

$$\phi(x) = \left[-\frac{1}{2}\frac{\partial^2}{\partial x^2}\right]\psi(x) = -\frac{1}{2}\psi''(x) \tag{2.16}$$

This term requires a bit more work because of the second derivative. We will massage the contiuous version a bit *before* discretizing.

First we proceed by Taylor expanding $\psi(x \pm \Delta x)$:

$$\psi(x + \Delta x) \approx \psi(x) + \frac{\Delta x}{1!}\psi'(x) + \frac{\Delta x^2}{2!}\psi''(x) + \frac{\Delta x^3}{3!}\psi'''(x) + O(\Delta x^4) \tag{2.17}$$

$$\psi(x - \Delta x) \approx \psi(x) - \frac{\Delta x}{1!}\psi'(x) + \frac{\Delta x^2}{2!}\psi''(x) - \frac{\Delta x^3}{3!}\psi'''(x) + O(\Delta x^4) \tag{2.18}$$

If we add two expressions, the odd expansion terms will cancel, and we can solve for $\psi''(x)$

$$\psi(x + \Delta x) + \psi(x - \Delta x) \approx 2\psi(x) + 2\frac{\Delta x^2}{2!}\psi''(x) + O(\Delta x^4) \Rightarrow \tag{2.19}$$

$$\psi''(x) \approx \frac{\psi(x - \Delta x) - 2\psi(x) + \psi(x + \Delta x)}{\Delta x^2} + O(\Delta x^4) \tag{2.20}$$

Now we are ready to discretize. By construction the following is true for all interior points $i \in [2:1:N-1]$

$$x_{i-1} = x_i - \Delta x \tag{2.21}$$

$$x_{i+1} = x_i + \Delta x \tag{2.22}$$

For the extreme points we need to choose boundary conditions.

[3]The analysis with $V(x,t)$ is identical since time is just a parameter. Here we omit it for clarity.

1. **Periodic boundaries**: let the grid 'loop' back on itself

$$x_N = x_1 - \Delta x \tag{2.23}$$
$$x_1 = x_N + \Delta x \tag{2.24}$$

2. **Hard boundaries**: since x_0 and x_{N+1} are not included in $[x_1 : \Delta x : x_N]$ the corresponding wave function values $\psi(x_0)$ and $\psi(x_{N+1})$ are not defined and must be discarded from equation 2.20.

Using equation 2.20 the discrete version of equation 2.12 with $A = T$ is

The Kinetic Energy Matrix

Define $\kappa \equiv -(2\Delta x^2)^{-1}$.

$$\vec{\phi} = \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_i \\ \vdots \\ \phi_N \end{pmatrix} \approx \kappa \begin{pmatrix} \delta \cdot \psi_N - 2\psi_1 + \psi_2 \\ \vdots \\ \psi_{i+1} - 2\psi_i + \psi_{i-1} \\ \vdots \\ \delta \cdot \psi_1 - 2\psi_N + \psi_{N-1} \end{pmatrix} = \kappa \begin{bmatrix} -2 & 1 & \dots & 0 & \delta \\ 1 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & -2 & \ddots & \vdots \\ 0 & & \ddots & \ddots & 1 \\ \delta & 0 & \dots & 1 & -2 \end{bmatrix} \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_i \\ \vdots \\ \psi_N \end{pmatrix} \equiv \mathbf{T}\vec{\psi} \tag{2.25}$$

with boundary conditions determined by $\delta \equiv \begin{cases} 1 & \text{if periodic boundaries} \\ 0 & \text{if hard boundaries} \end{cases}$.

Therefore the kinetic energy operator turns (approximately) into a $N \times N$ matrix $\mathbf{T}$ with -2 on the diagonal, $+1$ on the first super- and sub diagonal, and δ in the corners (determining boundary conditions), all with a factor of κ.
It is easy to increase the approximation accuracy to $O(\Delta x^6)$ by keeping terms up to order 5 in the Taylor expansion. This results in different scaling and also values on the second super- and sub diagonal, but the procedure is the same.

T+V: We have expressed both operators T and V in their matrix form, so the matrix form of the Hamiltonian operator is just

$$\mathbf{H} = \mathbf{T} + \mathbf{V} \tag{2.26}$$

with the overall structure identical to $\mathbf{T}$, since $\mathbf{V}$ is diagonal.

We are now partially done with the discretization procedure. We still have to specify how the time evolution steps in equation 2.8 are calculated, but solving the time independent Schrödinger equation has been transformed from an algebraic task to a numerical one

$$\hat{H}\phi_n = E_n\phi_n \xrightarrow{discretization} \mathbf{H}\vec{\phi}_n = E_n\vec{\phi}_n \tag{2.27}$$

The act of solving such an eigenvalue problem is often refered to as *diagonalization*, and in Matlab it can be performed by calling

```
[eigenvectors, eigenvalues] = eig(H)
```

It is important to note that the potential could be arbitrarily complex as in equation 1.7 – we just need to evaluate the potential in our discrete points, construct $\mathbf{H} = \mathbf{T} + \mathbf{V}$, and let the computer diagonalize $\mathbf{H}$. Figure 1 shows the results of this operation.
In the case $V = V(x)$ we could also perform time evolution by equation 1.5 if we really wanted, but in the next section we will discover a much more general approach that does not even require us to diagonalize!

Be aware that the continuous notation will be mainly used in the remainder of the note. The discrete notation will only appear as an indicator of what actually is done numerically at the end of the day. It is surprisingly difficult to strike a good balance between the two.

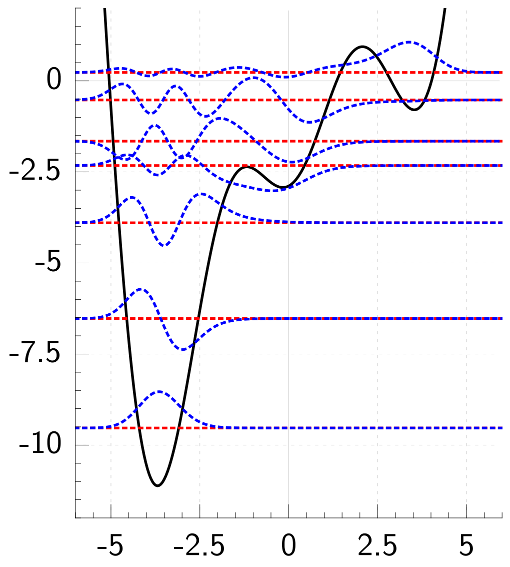


Figure 1: The first 5 eigenstates (blue) and corresponding eigenvalues (red) of H with the potential in equation 1.7, with parameters $a = b = \omega = \sigma = 1$, $x_0 = 0$, $A = -3$ at $t = 0.5$, and an x-grid from -6 to +6 in $N = 256$ points. The first 2-3 eigenstates resemble harmonic oscillator eigenstates, and subsequent eigenstates are highly non-trivial.

2.2 Time Evolution

Our final objective of the chapter is to describe the time evolution of our system. There are several ways to attack this problem, but under any circumstance it starts with the time dependent Schrödinger equation

$$i\dot{\psi}(x,t) = \hat{H}\psi(x,t) \tag{2.28}$$

We have already seen how to approximate $\partial_x^2\psi(x,t)$ when we calculated the kinetic energy, and it would seem reasonable that we could do something similar with $\dot{\psi}(x,t) = \partial_t\psi(x,t)$ on the left hand side. This is indeed the case, but we will pursue a different approach for at least 2 reasons

- It is easier to understand it from a physical point of view, and it is part of a larger framework.
- We will need it to derive the GRAPE algorithm in chapter 4

The price we need to pay is in establishing the machinery and concepts[4] , but it is well worth it for the above reasons.

2.2.1 The Time Evolution Operator

The observation we need to make is rather simple: the formal solution to the time dependent Schrödinger equation can be written

$$\psi(t) = e^{-i\hat{H}\cdot(t-t_0)}\psi(t_0) \tag{2.29}$$

where $\hat{H}$ is assumed independent of time for now. The definition of the *operator exponential* is

$$e^{t\hat{A}} \equiv \sum_{k=0}^{\infty} \frac{(t\hat{A})^k}{k!} = 1 + \frac{t\hat{A}}{1!} + \frac{t^2\hat{A}^2}{2!} + \frac{t^3\hat{A}^3}{3!} + \dots \tag{2.30}$$

The exponential $e^{-i\hat{H}\cdot(t-t_0)}$ is an operator exponential that is hard to calculate in general, and this is something we need to address later on. This particular exponential construct is so important we give it its

[4]A completely formal derivation from first principles is out of the scope of this note, but [2, chap. 2] is a good read on the subject

own symbol

$$\hat{\mathcal{U}}(t,t_0) = e^{-i\hat{H}\cdot(t-t_0)} \tag{2.31}$$

Let us describe equation 2.29 in words:

The operator $\hat{\mathcal{U}}(t,t_0)$ evolves the state ψ from $t_0 \to t$

It is for this reason $\hat{\mathcal{U}}(t,t_0)$ is referred to as the time evolution operator. In particular for the evolution $t \to t+\Delta t$ we obtain

$$\hat{\mathcal{U}} \equiv \hat{\mathcal{U}}(t+\Delta t, t) = e^{-i\hat{H}\Delta t} \tag{2.32}$$

A nice way to think about the operator is that it 'lives' in between two time steps.

An important property of $\hat{\mathcal{U}}$ is unitarity

$$1 = \hat{\mathcal{U}}^{\dagger}\hat{\mathcal{U}}$$

This allows us to think of $\hat{\mathcal{U}}^{\dagger} = e^{+iH\Delta t}$ as a backward propagation

$$\psi(t) = \left(\hat{\mathcal{U}}^{\dagger}\hat{\mathcal{U}}\right)\psi(t) = \hat{\mathcal{U}}^{\dagger}\left(\hat{\mathcal{U}}\psi(t)\right) = \hat{\mathcal{U}}^{\dagger}\psi(t+\Delta t)$$

Suppose we wanted to evolve from $t_0 = 0$ to some final time $T = (M-1)\Delta t$. That just amounts to $(M-1)$ applications of $\hat{\mathcal{U}}$

$$\hat{\mathcal{U}}(T,0) = \underbrace{\hat{\mathcal{U}}\cdot\hat{\mathcal{U}}\cdot\ldots\cdot\hat{\mathcal{U}}}_{M-1} = \prod_{j=1}^{M-1}\hat{\mathcal{U}} \tag{2.33}$$

We see that taking a lot of smaller, discrete steps is the same as taking one big step. The discretization in time emerged almost by itself! These results are analytically exact, but it was derived for time independent H only.

Fortunately we can generalize to time dependent H by approximating $H(t_j)$ to be constant over each time slice $[t_j; t_j+\Delta t]$.

The Time Evolution Operator

$$\hat{\mathcal{U}}(t_j) \equiv \hat{\mathcal{U}}(t_j+\Delta t, t_j) = e^{-i\hat{H}(t_j)\Delta t}, \tag{2.34}$$

$$\hat{\mathcal{U}}(T,0) = \underbrace{\hat{\mathcal{U}}(t_M)\cdot\hat{\mathcal{U}}(t_{M-1})\cdot\ldots\cdot\hat{\mathcal{U}}(t_1)}_{M} = \prod_{j=1}^{M}\hat{\mathcal{U}}(t_j) \tag{2.35}$$

$$1 = \hat{\mathcal{U}}^{\dagger}\hat{\mathcal{U}}, \qquad \text{(unitarity)} \tag{2.36}$$

$$\psi(T) = \prod_{j=1}^{M}\hat{\mathcal{U}}(t_j)\psi(0), \qquad \text{(forward propagation)} \tag{2.37}$$

$$\psi(0) = \prod_{j=M}^{1}\hat{\mathcal{U}}^{\dagger}(t_j)\psi(T), \qquad \text{(backward propagation)} \tag{2.38}$$

for sufficiently small Δt

The very last piece of the puzzle is to actually numerically apply $\hat{\mathcal{U}}(t_j)$ to $\psi(t_j)$. In our discretized notation this amounts to

$$\vec{\psi}(t_{j+1}) = \mathbf{U}(t_j)\vec{\psi}(t_j) = e^{-i\mathbf{H}(t_j)\Delta t}\vec{\psi}(t_j) \tag{2.39}$$

In Matlab the matrix exponential could be achieved by the line

```
U = expm(-1i*H*dt)
```

but this direct exponentiation is generally a very costly operation for the $N \times N$ matrices we are considering. The exception to this is when the matrix is diagonal. Then the matrix exponential is simply the exponentiation of each diagonal entry. This does not help us much at the moment, because **H** was certainly not diagonal, because of the kinetic energy matrix. We have to do something smarter. That something is called the *split-step method*, but to fully appreciate it we need to first discuss representations.

2.2.2 A Short Digression on Representations

If we could somehow make both matrices diagonal, we are in business. What made $\mathbf{V}(x)$ diagonal in the first place? The answer has to do with how we represented our states. This happens to be an extremely important topic in quantum mechanics [2, chap. 1]. For the sake of brevity we will only illuminate the main ideas here. If this is your first introduction to these concepts, do not worry if you do not understand it in full detail.

When we write the wave function $\psi = \psi(x)$ we are describing the state of the system in the so-called $\hat{x}$-basis representation. This is the most intuitive representation in the sense that it assigns a probability amplitude for finding the particle at each point in space. But we could also have described our state of the system in e.g. the $\hat{p}$-basis representation, the momentum-representation, in which case it is customary to write the wave function as $\phi = \phi(p)$. Now the wave function assigns a probability amplitudes of finding the particle with momentum p. Note the very distinct difference between *state* and *wave function*. A wave function is something you get when you decide on a particular basis to describe your state in.

It should be very obvious that $\psi(x) \neq \phi(p)$. In fact it can be shown that the 2 representations are connected by Fourier transforms

$$\phi(p) = \mathcal{F}[\psi(x)] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-ipx'} \psi(x') dx' \tag{2.40}$$

$$\psi(x) = \mathcal{F}^{-1}[\phi(p)] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{ip'x} \phi(p') dp' \tag{2.41}$$

After discretization, these operations can be cheaply performed using the discrete Fast Fourier Transform (FFT) algorithm.

Choosing a particular basis does not only affect wave functions, but also operators. Whenever we have written

$$\hat{H} = \hat{T} + \hat{V} = \frac{\hat{p}^2}{2} + \hat{V} = -\frac{1}{2}\frac{\partial^2}{\partial_x^2} + V(x,t) \tag{2.42}$$

we have (maybe unknowingly) specified that we are working in the $\hat{x}$-basis in the last equality, as the following table suggests

$\hat{x}$-basis	**$\hat{p}$-basis**
$\hat{x} = x$	$\hat{x} = i\partial_p$
$\hat{p} = -i\partial_x$	$\hat{p} = p$
$\hat{A}(\hat{x}) = A(x)$	$\hat{A}(\hat{p}) = A(p)$

A fundamental property of the $\hat{x}$-operator is that it is diagonal in the $\hat{x}$-basis by construction. Any $\hat{A} = \hat{A}(\hat{x})$ can also be shown to be diagonal. This exactly ensures that $\hat{V}(\hat{x},t) = V(x,t)$ is diagonal in the $\hat{x}$-representation, because time is not an operator.

By the same token the $\hat{p}$-operator is diagonal in the $\hat{p}$-basis by construction. This means that $\hat{T} = \frac{\hat{p}^2}{2}$ is also diagonal in this representation. Sometimes we instead prefer to write $\hat{T} = \frac{\hat{k}^2}{2}$ using $\hat{p} = \hbar\hat{k} = \hat{k}$.

Now we are very nearly done. The last thing we have to do is to make an approximation of the operator exponential.

2.2.3 The Split-Step Method

Remember that the goal is to calculate the operator exponential

$$\hat{\mathcal{U}} = e^{-i\hat{H}\Delta t} = e^{-i(\hat{T}+\hat{V})\Delta t} \tag{2.43}$$

in a smart way. We just showed that $\hat{V}$ is diagonal in the $\hat{x}$-basis, $\hat{T}$ is diagonal in the $\hat{p}$-basis, so the matrix exponentials of these are also diagonal *in their respective bases*. In addition we can move between the basis representations using Fourier transforms.

Unfortunately $e^{-i(\hat{T}+\hat{V})\Delta t} \neq e^{-i\hat{T}\Delta t}e^{i\hat{V}\Delta t}$ because $\hat{T}$ and $\hat{V}$ does not commute, but it is possible to show [3, chap. 26] that the time step can be split (hence the name of the method) in the following way

$$e^{-i(\hat{T}+\hat{V})\Delta t} \approx e^{-i\frac{\hat{V}}{2}\Delta t}e^{-i\hat{T}\Delta t}e^{-i\frac{\hat{V}}{2}\Delta t} + O(\Delta t^3) \tag{2.44}$$

Now we just have to combine everything. Starting from the current wave function $\vec{\psi}(t_j)$ we can calculate the wave function at the next time step $\vec{\psi}(t_{j+1})$

Split-Step Method

1. Calculate $e^{-i\frac{\hat{V}(t_j)}{2}\Delta t}$ in the $\hat{x}$-basis and apply it to $\psi(x,t_j)$
2. Fourier transform $\psi(x,t_j)$ to obtain $\phi(p,t_j)$ in the $\hat{p}$-basis
3. Calculate $e^{-i\hat{T}\Delta t}$ in the $\hat{p}$-basis and apply it to $\phi(p,t_j)$
4. Inverse Fourier transform $\phi(p,t_j)$ to obtain $\psi(x,t_j)$ in the $\hat{x}$-basis
5. Calculate $e^{-i\frac{\hat{V}(t_j)}{2}\Delta t}$ in the $\hat{x}$-basis and apply it to $\psi(x,t_j)$

or written explicitly in a single line using the discretized matrix/vector form

$$\vec{\psi}(t_{j+1}) = e^{-i\frac{\mathbf{V}(t_j)}{2}\Delta t}[\mathcal{F}^{-1}[e^{-i\mathbf{T}\Delta t}[\mathcal{F}[e^{-i\frac{\mathbf{V}(t_j)}{2}\Delta t}[\vec{\psi}(t_j)]]]]] \tag{2.45}$$

It may look daunting, but every single operation is simple and cheap in itself. There are some improvements that can be made to this approximation, but they will not be discussed here. As a technical aside, another effect of the discretization of x on a regular grid is that the wavenumbers k also become discretized. Without going into details the wavenumbers needed are (in Matlab notation)

```
kvec = 2*pi/L*[0:N/2−1 −N/2:−1];
```

This concludes the chapter. Given any arbitrary initial wave function $\psi_{initial} = \psi(x,0)$ we can evolve it in any arbitrary potential $V(x,t)$ to some final wave function $\psi(x,T)$ in $(M-1)$ time evolutions of length Δt using the split-step method. We could write this process succinctly as

The Time Evolution Protocol

choosing $\psi(0)$, $V(x,t)$, and $T = (M-1)\Delta t$ produces

$$\psi(0) \xrightarrow[V(x,t)]{T} \psi(T) \tag{2.46}$$

in $(M-1)$ applications of the split-step method

As an example, figure 2 plots the probability distribution $|\psi(x,t)|^2$ for a state $\psi_{initial} = e^{-0.5x^2}$ being evolved from $t=0$ to $T=3$ in the potential $V(x,t)$ from equation 1.7.

3 The Transport Problem

In the previous chapter we showed how to evolve wave functions in arbitrarily complicated time dependent potentials. The goal of this chapter is to make good use of this tool, and introduce concepts from Quantum

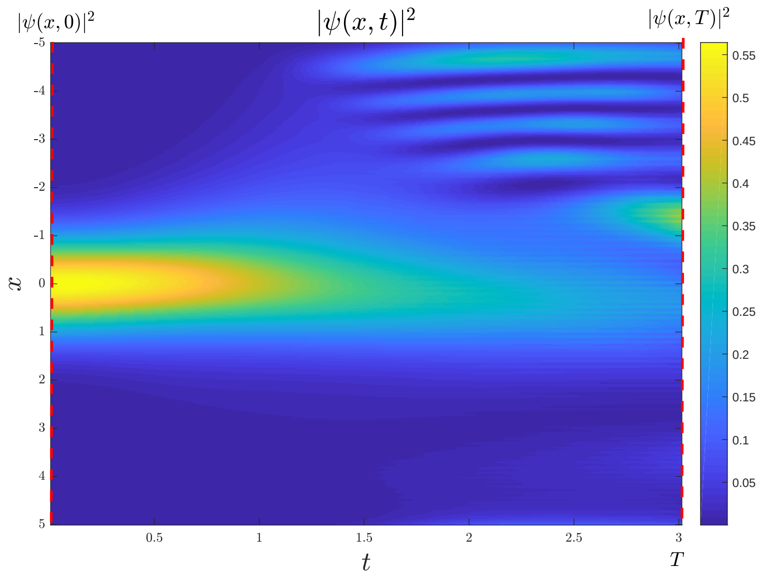


Figure 2: Time evolution of $\psi_{initial} = e^{-0.5x^2}$ in potential from equation 1.7 with $a = b = \omega = \sigma = 1$ from $t = 0$ to $T = 3$ with $\Delta t = 0.01$.

Optimal Control Theory (QOCT) to set the scene for the next, and final, chapter. Specifically we will define and work with the *Transport* problem.

3.1 The State Transfer Control Problem

Equation 3.2 tells us that for some fixed initial state $\psi_{initial}$ and fixed duration T, each choice of $V(x,t)$ produces *some* final state $\psi(T)$. Suppose we were interested in ending up in some target state ψ_{target}. That is, we want to find $V(x,t)$, that realizes $\psi(T) = \psi_{target}$ after propagation. To measure how well the final state $\psi(T)$ obtained for any particular $V(x,t)$ performs in this task we can look at the *fidelity*

$$\mathcal{F} = |\langle\psi_{target}|\psi(T)\rangle|^2 = |\int_{-\infty}^{\infty} \psi^*_{target}\psi(T)dx|^2 \tag{3.1}$$

The better the match, the higher the $\mathcal{F}$. The *optimal* solution is a perfect match[5] $\psi(T) = \psi_{target}$ where $\mathcal{F} = 1$. In other words we want to find a $V(x,t)$ that maximizes fidelity $\mathcal{F}$.

This is a highly complex optimization problem. For one, there are infinitely many potentials to choose from. Another issue is that there might not even *exist* an optimal solution. The former issue is something we can tackle to some extent by parameterizing the potentials by a *control function*[6] $u(t)$, i.e. $V(x,t,u(t))$. We then restrict ourselves to only being able to manipulate the potential through this $u(t)$. This also turns out to often be reasonable from a physical point of view. Dreaming up arbitrary potentials is easy enough in theory, less so is actually creating them in an experimental setting. The restrictions imposed by experiment is sometimes even a net positive since the control function $u(t)$ may be a natural quantity of the system. As an example, the control function $u(t)$ in the *Transport* problem is simply the position of an optical tweezer, as we will discuss shortly.

Since we established V can now only be changed through $u(t)$, and that we want to maximize the fidelity $\mathcal{F} = |\langle\psi_{target}|\psi(T)\rangle|^2$, we may state the optimal control problem as

[5]Up to an overall-phase
[6]It is possible to have more than one control function

The State Transfer Optimal Control Problem

find $u(t)$ realizing

$$\psi(0) \xrightarrow[u(t)]{T} \psi(T) \tag{3.2}$$

such that $\mathcal{F}[u(t)] = |\langle\psi_{target}|\psi(T)\rangle|^2$ is maximized

Understanding the box above is very important. To gain some practical insight we now discuss the *Transport* problem in detail, using the terminology we have just developed.

3.2 State Transfer: The *Transport* problem

The *Transport* problem deals with moving an atom from A to B in some fixed duration time T. In this example we trap the atom using an intense and tightly focused laser to generate a gaussian potential $V(x,t)$ with depth a, width σ, centered on $x_0(t)$

$$V(x,t) = -a \cdot e^{-\frac{(x-x_0(t))^2}{2\sigma^2}} \tag{3.3}$$

This is called an *optical tweezer*. At $t = 0$ the tweezer is located at $x_0(t = 0) = A$, and the atom is initially at rest in the ground state of the tweezer. At time $t = T$ the tweezer has been moved to $x_0(t = T) = B$, the target wave function we aim for is the ground state of this moved tweezer. As previously alluded to, a natural control function in this problem is the tweezer position $u(t) = x_0(t)$. Any control function starting in A and ending in B is admissible. The following box summarizes and illustrates the problem.

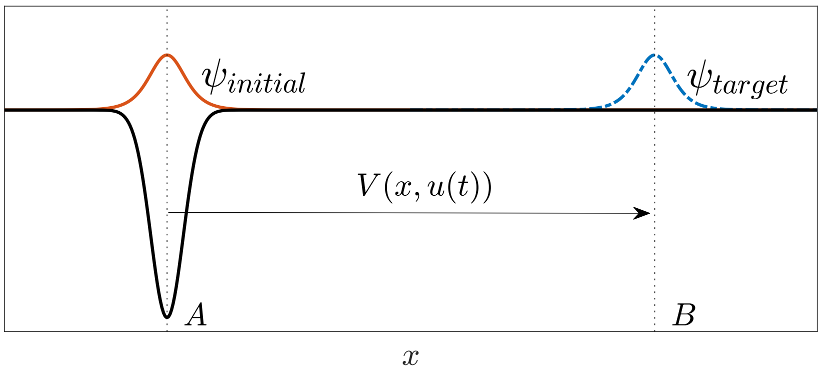
The Transport Problem

$$V(x,u(t)) = -a \cdot e^{-\frac{(x-u(t))^2}{\sigma^2}} \tag{3.4}$$

$$u(t): \text{ tweezer position starting in } A \text{ and ending in } B \tag{3.5}$$

$$\psi_{initial}: \text{ ground state of tweezer centered on } A \tag{3.6}$$

$$\psi_{target}: \text{ ground state of tweezer centered on } B \tag{3.7}$$



Now we need to find the $u(t)$ that maximizes $\mathcal{F}$. As an example we could try the control function

$$u(t) = A + (1 - \frac{A}{B})\sin(\frac{t}{T}\frac{\pi}{2})B \tag{3.8}$$

This control is valid because $u(0) = A$, $u(T) = B$. The transfer sequence for this particular control is plotted in figure 3.
The three dots correspond to pairs of $(t, u(t))$ at $t = 0$, $t = T/2$, and $t = T$.
Figure 3 a) shows the control as a function of time $u(t)$.

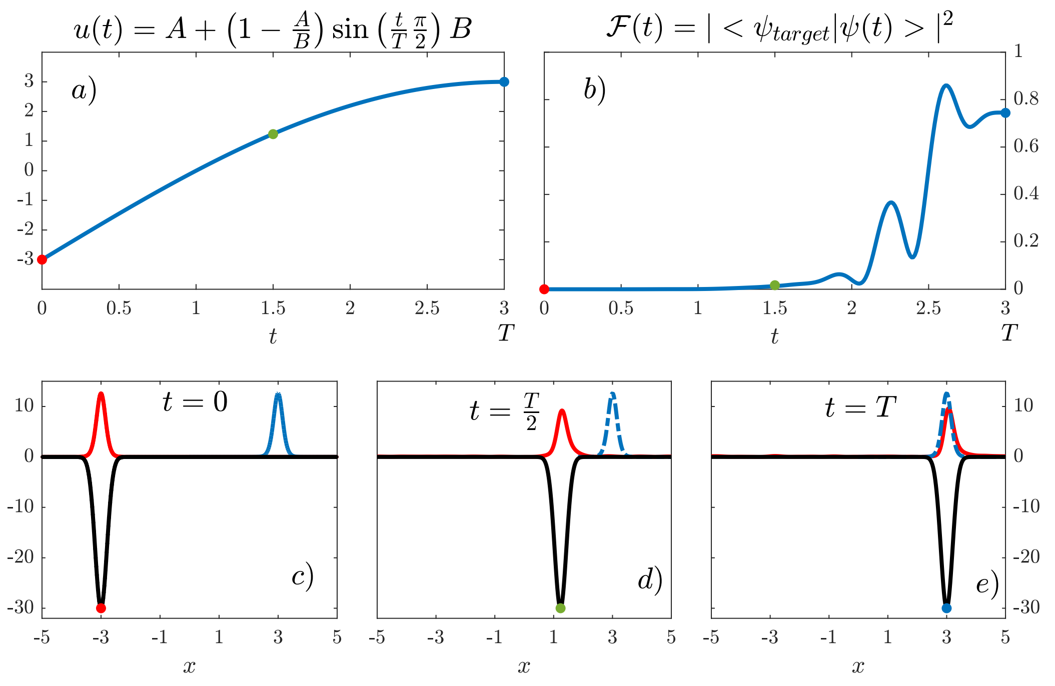


Figure 3: a) u(t) b) $\mathcal{F}(t)$ c-e) System at $t = 0$, $t = \frac{T}{2}$, and $t = T$.

Figure 3 b) shows the instantaneous fidelity $\mathcal{F}(t)$
Figure 3 c-e) shows $|\psi(x,t)|^2$ (red), $V(x,t)$ (black) at three points in time, and also shows $|\psi_{target}(x,t)|^2$ (blue).

The final fidelity for this control is $\mathcal{F} = 0.7452$. While this is not terrible, it is still very far from 1! So what can we do now? Say we are not allowed to change T, maybe for some experimental reason or something like that. Do we then just concoct control functions $u_1(t), u_2(t), \ldots, u_N(t)$ until we get $\mathcal{F}[u_N] = 1$? It seems like a fool's errand, because there are infinitely many $u(t)$ to choose from, and the one(s) giving perfect fidelity is probably not straightforward to write down in an algebraic form. In practice we can relax the criterion a bit, such that $\mathcal{F} \geq 0.999$ is ok. But currently this does not help us much for exactly the same reasons as before.

Let's consider again the result $\mathcal{F} = 0.7452$. Given a bit of thought, one can argue that this particular control is actually a part of an extremely restricted subset of all possible $u(t)$ that produces anything remotely close to 1. We can also see from figure 3 b) that the maximum fidelity is $\mathcal{F} \approx 0.83$ at about $t = 2.7$. Maybe we could somehow adjust our control a bit by adding a correction $\eta(t)$ around $t = 2.7$ that stops the fidelity from dipping down afterwards. Suppose we by *extreme luck* found a correction such that

$$\tilde{u}(t) = u(t) + \eta(t) \tag{3.9}$$
$$\mathcal{F}[\tilde{u}(t)] = 0.83 \tag{3.10}$$

Great, this is a huge improvement! But we still have a long way to go, and now it's hard to identify how to improve the control. Maybe we could just try to add some other correction $\tilde{\eta}(t)$ around $t = T/2$ and evaluate the fidelity of this control:

$$\overline{u}(t) = \tilde{u}(t) + \tilde{\eta}(t) \tag{3.11}$$
$$\mathcal{F}[\overline{u}(t)] = 0.64 \tag{3.12}$$

Apparently this correction made the fidelity even worse than the original one. By modifying the control at $t = T/2$ we modified the entire following history, such that the first $\tilde{\eta}(t)$ is no longer the correction we needed in the first place!

This line of thoughts illustrate that it might be a great idea to modify "good" controls by adding corrections. What the experiment also illustrates is that doing it by hand in reasonable time is impossible. What we need is a fast, systematic way to determine how the corrections should look like to improve our solution. In the next chapter we will attack this problem head on using *Optimal Control Theory*.

4 Optimal Control Theory

Optimization and Optimal Control Theory are two very well developed fields, and the surrounding theory is quite extensive and advanced. In this note we will only discuss what is absolutely required. We will introduce the GRAPE algorithm, which is a method to increase the fidelity by improving our control iteratively. Once we have done that we will discuss the Krotov algorithm, which for our purpose is simply a more sophisticated and advanced form of the GRAPE algorithm.

4.1 Optimizing a real function $f(\vec{x})$

Suppose we want to find the maximum of the function $f(\vec{x})$ shown in figure 4 a). We can obviously find it just by inspecting the plot, but this generally not the case. Instead we will describe a method that will find a maximum by iteratively improving an initial guess. Our only assumption will be that we can calculate the gradient $\vec{\nabla} f(\vec{x})$ in some way.

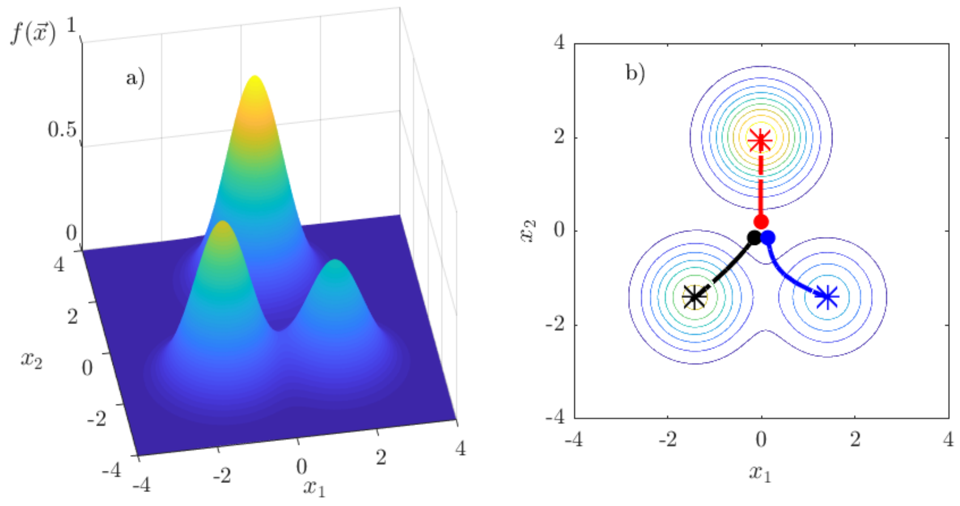


Figure 4: a) Example function to be maximized b) 20 Steepest ascent iterations with constant stepsize $\alpha = 1$, starting in the solid dots, and ending in the stars. Each initial point end up in a different maximum.

The following analogy will sketch the main idea: Imagine parachuting onto a smooth mountain, and your goal is to reach the top. Unfortunately the mountain is shrouded in a thick fog. The fog is so thick that you can only see the ground in a 1 foot radius around you. With only this local information available, how do you proceed to climb the mountain? The easiest way is this: identify the direction of steepest increase, take a step in that direction, then repeat the procedure until you have reached the top. If the mountain looked like figure 4 a), the maximum you find will depend a lot on where you intially landed. Figure 4 b) shows how 3 close, but different, initial points (solid dots) leads to very different maxima (stars).

Now we translate this analogy to a mathematical form[4]. We start out at point $\vec{x}_1$, decide on a direction $\vec{p}_1$, take a step of length α in that direction, which gets us to the point $\vec{x}_2$, and repeat the procedure until converged. The direction of *steepest* local increase at $\vec{x}_k$ is the gradient at $\vec{p}_k = \vec{\nabla} f(\vec{x}_k)$. This is the simplest gradient-based method in optimization, that for obvious reasons is called the *method of steepest ascent*. Additionally we can also find the best $\alpha = \alpha_k$ at each iteration. This is called a *linesearch*, because we are restricted to only move on the line defined by the gradient. This is then a 1D optimization problem for the stepsize α_k. For our purposes it is enough to known that this can be done.

The Method of Steepest Ascent with Linesearch

choose an inital point $\vec{x}_1$

At iteration k: calculate local gradient $\vec{\nabla} f(x_k)$ and find best stepsize α_k

$$\vec{x}_{k+1} = \vec{x}_k + \alpha_k \vec{\nabla} f(x_k) \tag{4.1}$$

such that

$$f(x_{k+1}) \geq f(x_k) \tag{4.2}$$

the gradient direction at $\vec{x}_k$ is the direction of steepest increase of f near $\vec{x}_k$

The task of finding a good intial point is pervasive in optimization theory. Often the best we can do is give an educated guess, based on our knowledge of the problem at hand.

4.2 The GRAPE Algorithm

Gradient **A**scent **P**ulse **E**ngineering (GRAPE) is the simplest optimal control algorithm. It is simply the steepest ascent method applied to our control functions. To proceed we need to calculate the gradient of the fidelity with respect to the control function $u(t)$. Notice however that the steepest ascent method described above applies to functions $f(\vec{x})$ accepting *vectors*, while our fidelity $\mathcal{F}[u(t)]$ accepts *functions*. Such a function of a function is called a *functional*, and the derivatives of functionals are called *functional derivatives*.

Luckily there is a simple way to calculate the gradient without using functional derivatives. When we discretize our control function $u(t)$ on our t-grid the fidelity becomes simply $\mathcal{F}[\vec{u}]$, i.e. a normal function of $\vec{u}$ like the $f(\vec{x})$ example above. We can also write the fidelity in terms of the time evolution operators that at t_j has the form $\hat{\mathcal{U}}_j = \hat{\mathcal{U}}(u(t_j))$:

The Discretized Control and the Resulting Fidelity

$$\vec{u} = \big(u(t_1), \ldots, u(t_j), \ldots, u(t_M)\big) = \big(u_1, \ldots, u_j, \ldots, u_M\big) \tag{4.3}$$

$$j \in [1:1:\text{length}(\vec{t})] \tag{4.4}$$

using equation 2.38 the fidelity becomes

$$\hat{\mathcal{U}}_j \equiv \hat{\mathcal{U}}(u(t_j)) = e^{-i(\hat{T}+\hat{V}(u(t_j)))\Delta t} \tag{4.5}$$

$$F[\vec{u}] = |\langle\psi_{target}|\psi(T)\rangle|^2 = |\langle\psi_{target}|\prod_{j=1}^{M}\hat{\mathcal{U}}_j|\psi(0)\rangle|^2 \tag{4.6}$$

What remains is to calculate the gradient of the fidelity in equation 4.6

$$\vec{\nabla}_{\vec{u}} F(\vec{u}) = \begin{pmatrix} \frac{\partial}{\partial u_1} \\ \vdots \\ \frac{\partial}{\partial u_j} \\ \vdots \\ \frac{\partial}{\partial u_M} \end{pmatrix} F[\vec{u}]. \qquad j \in [1:1:\text{length}(\vec{t})] \tag{4.7}$$

In the following we provide a detailed step-by-step calculation of $\frac{\partial}{\partial u_j} F[\vec{u}]$.

For reference we use the following definitions and identities:

(1) Define $c \equiv \langle\psi_{target}|\psi(T)\rangle = \langle\psi_{target}|\prod_{j'=1}^{M-1}\hat{\mathcal{U}}_{j'}|\psi(0)\rangle \Rightarrow F = |c|^2 = c^*c$

(2) If $\kappa = a + ib \Rightarrow \kappa^* + \kappa = (a - ib) + (a + ib) = 2a = 2\mathcal{R}(\kappa)$

(3) Define $\langle\chi(T)| \equiv -i\,\langle\psi(T)|\psi_{target}\rangle \cdot \langle\psi_{target}|$

(4) Forward evolution: $|\psi(t_j)\rangle = \prod_{j'=1}^{j}\hat{\mathcal{U}}_{j'}\,|\psi(0)\rangle$

(5) Backward evolution: $|\psi(t_j)\rangle = \prod_{j'=M-1}^{j}\hat{\mathcal{U}}_{j'}^{\dagger}\,|\psi(T)\rangle$

(6) Conjugated backward evolution:

$$\langle\chi(t_j)| = \Big(|\chi(t_j)\rangle\Big)^{\dagger} = \left(\prod_{j'=M-1}^{j}\hat{\mathcal{U}}_{j'}^{\dagger}\,|\chi(T)\rangle\right)^{\dagger} = \langle\chi(T)|\prod_{j'=j}^{M-1}\hat{\mathcal{U}}_{j'}$$

The identities are annotated as "$\overset{(i)}{=}$" where used in the derivation
We start with

$$\frac{\partial}{\partial u_j}F[\vec{u}] = \frac{\partial}{\partial u_j}|\,\langle\psi_{target}|\prod_{j'=1}^{M-1}\hat{\mathcal{U}}_{j'}|\psi(0)\rangle\,|^2 \overset{(1)}{=} \frac{\partial}{\partial u_j}(c^*c) \tag{4.8}$$

$$= \frac{\partial c^*}{\partial u_j}c + c^*\frac{\partial c}{\partial u_j} \overset{(2)}{=} 2\mathcal{R}(c^*\frac{\partial c}{\partial u_j}) \tag{4.9}$$

Now we focus on $\frac{\partial c}{\partial u_j}$:

$$\frac{\partial c}{\partial u_j} = \frac{\partial}{\partial u_j}\,\langle\psi_{target}|\prod_{j'=1}^{M-1}\hat{\mathcal{U}}_{j'}\,|\psi(0)\rangle \tag{4.10}$$

$$= \frac{\partial}{\partial u_j}\,\langle\psi_{target}|\hat{\mathcal{U}}_{M-1}\ldots\hat{\mathcal{U}}_{j+1}\hat{\mathcal{U}}_{j}\hat{\mathcal{U}}_{j-1}\ldots\hat{\mathcal{U}}_{1}\,|\psi(0)\rangle \tag{4.11}$$

$$= \langle\psi_{target}|\hat{\mathcal{U}}_{M-1}\ldots\hat{\mathcal{U}}_{j+1}\frac{\partial\hat{\mathcal{U}}_j}{\partial u_j}\hat{\mathcal{U}}_{j-1}\ldots\hat{\mathcal{U}}_{1}\,|\psi(0)\rangle \tag{4.12}$$

For $\frac{\partial\hat{\mathcal{U}}_j}{\partial u_j}$ we have[7]

$$\frac{\partial\hat{\mathcal{U}}_j}{\partial u_j} = \frac{\partial}{\partial u_j}\left(e^{-i(\hat{T}+\hat{V}(u_j))\Delta t}\right) = \frac{\partial}{\partial u_j}\left(-i(\hat{T}+\hat{V}(u_j))\Delta t\right)\hat{\mathcal{U}}_j \tag{4.13}$$

$$= -i\Delta t\frac{\partial\hat{V}(u_j)}{\partial u_j}\hat{\mathcal{U}}_j \tag{4.14}$$

Inserting equation 4.14 in equation 4.12 above yields

$$\frac{\partial c}{\partial u_j} = -i\Delta t\,\langle\psi_{target}|\hat{\mathcal{U}}_{M-1}\ldots\hat{\mathcal{U}}_{j+1}\frac{\partial\hat{V}(u_j)}{\partial u_j}\hat{\mathcal{U}}_{j}\hat{\mathcal{U}}_{j-1}\ldots\hat{\mathcal{U}}_{1}\,|\psi(0)\rangle \tag{4.15}$$

$$= -i\Delta t\,\langle\psi_{target}|\prod_{j'=j+1}^{M-1}\hat{\mathcal{U}}_{j'}\frac{\partial\hat{V}(u_j)}{\partial u_j}\prod_{j'=1}^{j}\hat{\mathcal{U}}_{j'}|\psi(0)\rangle \tag{4.16}$$

[7] In the second equation we assume the exponent and the derivative of the exponent commute. In reality it does not, but we ignore the error gained in this approximation.

Multiplying equation 4.16 with c^* gives

$$c^* \frac{\partial c}{\partial u_j} = \Delta t\Big(-i \langle \psi(T) | \psi_{target} \rangle \cdot \langle \psi_{target} | \Big) \prod_{j'=j+1}^{M-1} \hat{\mathcal{U}}_{j'} \frac{\partial \hat{V}(u_j)}{\partial u_j} \prod_{j'=1}^{j} \hat{\mathcal{U}}_{j'} |\psi(0)\rangle$$

$$\overset{(3)}{=} \Delta t \underbrace{\Big(\langle \chi(T) | \prod_{j'=j+1}^{M-1} \hat{\mathcal{U}}_{j'} \Big)}_{\langle \chi(t_j)|} \frac{\partial \hat{V}(u_j)}{\partial u_j} \underbrace{\Big(\prod_{j'=1}^{j} \hat{\mathcal{U}}_{j'} |\psi(0)\rangle \Big)}_{|\psi(t_j)\rangle} \tag{4.17}$$

$$\overset{(4)+(5)+(6)}{=} \Delta t \langle \chi(t_j) | \frac{\partial \hat{V}(u_j)}{\partial u_j} |\psi(t_j)\rangle \tag{4.18}$$

Finally we just reinsert equation 4.18 into equation 4.9 to obtain the final result

$$\frac{\partial}{\partial u_j} F[\vec{u}] = 2\mathcal{R}(c^* \frac{\partial c}{\partial u_j}) = 2\Delta t \mathcal{R}\Big(\langle \chi(t_j) | \frac{\partial \hat{V}(u_j)}{\partial u_j} |\psi(t_j)\rangle \Big) \tag{4.19}$$

This is the gradient for the control $\vec{u}$ at time t_j. There are few simple steps to computing the full gradient $\nabla_{\vec{u}} \mathcal{F}[\vec{u}]$:

1. Starting from $\psi(0) = \psi_{initial}$, calculate all $\psi(t_j)$ by forward propagation over the current $\vec{u}$
2. Starting from $\chi(T) = i \langle \psi_{target} | \psi(T) \rangle \cdot \psi_{target}$, calculate all $\chi(t_j)$ by backward propagation over the current $\vec{u}$.
3. Calculate $\frac{\partial \hat{V}}{\partial u}$ analytically and evaluate it for every element in the current $\vec{u}$

For the *Transport* problem the derivative of the potential is easily calculated in the x-representation

$$\frac{\partial}{\partial u} V(x,u) = \frac{\partial}{\partial u} \left(-a \cdot e^{-\frac{(x-u)^2}{2\sigma^2}} \right) = \frac{(x-u)}{\sigma^2} \cdot V(x,u) \tag{4.20}$$

We have found a way to calculate the gradient efficiently by using the steps above. The rest is just using steepest ascent with linesearch.

The GRAPE Algorithm with Linesearch

choose an inital control $\vec{u}_1$

At iteration k: calculate local gradient $\nabla_{\vec{u}}\mathcal{F}[\vec{u}_k]$ and find best stepsize α_k

$$\vec{u}_{k+1} = \vec{u}_k + \alpha_k \vec{\nabla}_{\vec{u}}\mathcal{F}[\vec{u}_k] \tag{4.21}$$

such that

$$\mathcal{F}[\vec{u}_{k+1}] \geq \mathcal{F}[\vec{u}_k] \tag{4.22}$$

where the gradient elements are calculated by

$$\frac{\partial}{\partial u_j} F[\vec{u}] = 2\Delta t \mathcal{R}\left(\langle \chi(t_j)| \frac{\partial V(u_j)}{\partial u_j} |\psi(t_j)\rangle \right) \tag{4.23}$$

with time evolution over the current control $\vec{u}$

$$\psi(t_j) = \prod_{j'=1}^{j} \hat{\mathcal{U}}_{j'}\psi(0) \qquad \text{(forward propagation)} \tag{4.24}$$

$$\chi(t_j) = \prod_{j'=M-1}^{j} \hat{\mathcal{U}}_{j'}^{\dagger}\chi(T) \qquad \text{(backward propagation)} \tag{4.25}$$

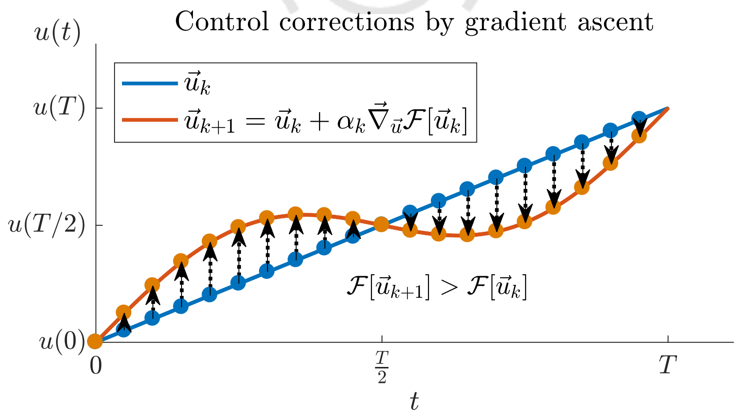
under the boundary conditions

$$u(0) = A \qquad \text{(fixed)} \tag{4.26}$$

$$u(T) = B \qquad \text{(fixed)} \tag{4.27}$$

$$\psi(0) = \psi_{initial} \qquad \text{(fixed)} \tag{4.28}$$

$$\chi(T) = i\,\langle \psi_{target}|\psi(T)\rangle\,\psi_{target} \qquad (\psi(T) \text{ produced by } \vec{u}) \tag{4.29}$$



We inherit the same problems as in the $f(\vec{x})$ case: choosing different inital control $\vec{u}_1$ can be difficult, and we are only guaranteed to climb to a local maximum. Compared to the 2 dimensional $f(\vec{x})$ example, this problem is a much larger M dimensional problem, because we optimize the control at each discrete point in time t_j. There is no general way to decide on the initial controls, other than using physical insight on the particular problem at hand.

In practice one often use several stopping criteria for the optimization algorithm because of limited computing time:

- $\mathcal{F}[u_k] > \mathcal{F}_{\text{threshold}}$ (usually around $0.9 - 0.999$)
- Iteration number k exceeds some maximum

- Total number of time propagations exceeds some maximum
- The norm of the gradient is too small
- The difference in fidelity between iteration $k+1$ and k is too small

4.3 Improving GRAPE

The GRAPE algorithm is the simplest case of an optimal control algorithm, and it was derived without use of the normally quite extensive optimal control framework. We now literally add some sophistication to the model. We start by slightly modifying the question we ask.

4.3.1 $J_{\mathcal{F}}$: Fidelity Cost

Instead of maximizing $\mathcal{F}$ we minimize $-\mathcal{F}$. The 2 problems are *completely* equivalent, but minimization is the usual strategy in optimization literature. Imagine flipping the sign of figure 4 a): the mountains become valleys. Instead of climbing up, we want to climb down. To do this we simply go in the opposite direction of the gradient. This is known as *steepest descent*.

For purely cosmetic reasons we also add a $+1$ and scale by a factor 0.5. We call this object a *cost*

$$J_{\mathcal{F}} = \frac{1}{2}\left(1-\mathcal{F}\right) = \frac{1}{2}\left(1 - |\langle\psi_{target}|\psi(T)\rangle|^2\right) \tag{4.30}$$

The way we intend to improve the sophistication of our model is to add several different costs together, each representing something we would like to take into account, into a total cost J. We then try to minimize this sum of costs. To ease notation we omit writing $\vec{u}_k$ as an argument in all the cost expressions.

4.3.2 J_{TDSE}: TDSE Cost

We know that ψ evolves according to the Schrödinger equation, and we want a cost that somehow reflect this. In the simple GRAPE derivation we explicitly inserted the time evolution operator to get equation 4.6, but here we introduce it as a cost instead

$$J_{TDSE} = \frac{1}{2}\int_0^T \left[\langle\chi(t)|\left(i\partial_t - \hat{H}\right)|\psi(t)\rangle + c.c.\right] dt \tag{4.31}$$

where the state $\chi(t)$ is called the *Lagrange multiplier*, and c.c denotes complex conjugation of the first term[8]. This cost forces $\psi(t)$ to obey the Schrödinger equation at all times[9].

4.3.3 J_γ: Regularization Cost

Suppose now that we for experimental reasons want our control to be smooth. Maybe we cannot physically change the position of our laser arbitrarily fast. We can define a cost that penalizes quickly varying controls

$$J_\gamma = \frac{\gamma}{2}\int_0^T \dot{u}(t)^2 dt \tag{4.32}$$

where γ is the *regularization* parameter. The larger (smaller) the γ, the less (more) the controls are allowed to vary change over time.

4.3.4 J: Total Cost

Now we have distinct cost terms, each serving a distinct purpose. We then write the *total cost* as

$$J = J_{\mathcal{F}} + J_{TDSE} + J_\gamma \tag{4.33}$$

This is then the cost that we minimize. Doing the functional derivatives on J results in a different gradient $\vec{\nabla}_{\vec{u}} J[\vec{u}_k]$ than before, but with the same equations of motion for ψ and χ. Instead of just trying to minimize

[8]The c.c. ensures that J_{TDSE} is real

[9]At this point we could do the functional derivatives wrt. $\psi(t)$, $\chi^*(t)$, and $u(t)$ to obtain exactly the same GRAPE algorithm we obtained in the last section

$J_{\mathcal{F}}$ we also have to compete with the regularization J_γ. The J_{TDSE} term is always zero, but it is required for calculating the equations of motion for ψ, χ, and the gradient. Finding suitable values for γ is usually more art than science, but usually lie within $10^{-1} - 10^{-4}$, or completely turned off at 0. The gradient will be stated without proof[5].

The Improved* GRAPE *Cost

minimize $J = J_{\mathcal{F}} + J_{TDSE} + J_\gamma$ where

$$J_{\mathcal{F}} = \frac{1}{2}(1-\mathcal{F}) = \frac{1}{2}\left(1 - |\langle\psi_{target}|\psi(T)\rangle|^2\right) \tag{4.34}$$

$$J_{TDSE} = \frac{1}{2}\int_0^T \langle\chi(t)|\left(i\partial_t - \hat{H}\right)|\psi(t)\rangle\, dt + c.c. \tag{4.35}$$

$$J_\gamma = \frac{\gamma}{2}\int_0^T \dot{u}(t)^2 dt \qquad \text{(regularization)} \tag{4.36}$$

with the gradient of this improved cost

$$\frac{\partial}{\partial u_j} F[\vec{u}] = \Delta t \mathcal{R}\left(\langle\chi(t_j)| \frac{\partial V(u_j)}{\partial u_j} |\psi(t_j)\rangle \right) - \gamma \ddot{u}_j \tag{4.37}$$

The algorithm is the same as previously, but uses *steepest descent*.

5 Units and non-dimensionalization

In this section I try to give a detailed explanation of what it means to say some thing like $\hbar = m = 1$.

The SI-unit system usually results in very small numerical values for quantum mechanical simulations because e.g. $\hbar = 1.05 \cdot 10^{-34}\text{m}^2\text{kg/s}$, $m_{electron} = 9.11 \cdot 10^{-31}\text{kg}$, and so on. This makes simulations impractical or infeasible due to for example the finite number precision inherent to computers. For this reason it is beneficial to rescale physical quantities into characteristic scales of the problem such that most values are of order unity. This is achieved using a process known as nondimensionalization where quantities in SI-units are written in product form e.g. a becomes

$$a = \mu_{[\text{a}]} \cdot \tilde{a} \tag{5.1}$$

where $\mu_{[\text{a}]}$ carries both the **dimension** of a and a **magnitude** while $\tilde{a}$ is a **non-dimensional** scaling value. This is done for all relevant quantities and substituted into the equations under consideration, which leaves a new set of working equations involving only the dimensionless scaling values. We can therefore also think of $\tilde{a}$ as **simulation values**. The choice of how to define a particular $\mu_{[\text{a}]}$ can be motivated by for example wanting to simplify the equations under consideration – for example obtaining equations with $\hbar = m = 1$.

This idea is more easily understood through example. The details can be fairly problem specific, but the general idea is the same. Usually we are interested quantities of length, time, and energy. Let L, T, E denote generic quanitites of these respective dimensions in SI-units and we get

$$L = \mu_{\text{length}} \cdot \tilde{L}, \quad T = \mu_{\text{time}} \cdot \tilde{T}, \quad E = \mu_{\text{energy}} \cdot \tilde{E} \tag{5.2}$$

How do we then choose $\mu_{[\text{a}]}$'s such that we get the usual Schrödinger equations, but with $\hbar = m = 1$? We can start by looking at the time-independent version,

$$E_n\psi_n = \hat{H}\psi_n = \left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V\right]\psi_n \tag{5.3}$$

Remembering x is a quantity with dimensions of length, we can substitute in the factored version $x =$

$\mu_{\text{length}} \cdot \tilde{x}$:

$$E_n\psi_n = \left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V\right]\psi_n = \left[-\frac{\hbar^2}{2m\mu_{\text{length}}^2}\frac{\partial^2}{\partial \tilde{x}^2} + V\right]\psi_n \tag{5.4}$$

$$= \frac{\hbar^2}{m\mu_{\text{length}}^2}\left[-\frac{1}{2}\frac{\partial^2}{\partial \tilde{x}^2} + \frac{m\mu_{\text{length}}^2}{\hbar^2}V\right]\psi \tag{5.5}$$

The front factor has dimensions of energy and we could conveniently choose $\mu_{\text{energy}} = \frac{\hbar^2}{m\mu_{\text{length}}^2}$. Note that the terms inside the parenthesis are dimensionless. Since V and E_n are quantities with dimensions of energy we also have $V = \mu_{\text{energy}} \cdot \tilde{V}$ and $E_n = \mu_{\text{energy}} \cdot \tilde{E}_n$. We may then rewrite as

$$E_n\psi_n = \mu_{\text{energy}} \cdot \left[-\frac{1}{2}\frac{\partial^2}{\partial \tilde{x}^2} + \tilde{V}\right]\psi \Rightarrow \tilde{E}_n\psi_n = \left[-\frac{1}{2}\frac{\partial^2}{\partial \tilde{x}^2} + \tilde{V}\right]\psi \equiv \hat{\tilde{H}}\psi \tag{5.6}$$

Now we have an equation of the exact same form as before, but with $\hbar = m = 1$.

Example: Eigen energies in the infinite square well

Let $V(x)$ be the infinite square well, i.e. zero potential for $x \in [0, l]$ and ∞ everywhere else. The eigen energies in SI-units are

$$E_n = \frac{n^2\pi^2\hbar^2}{2ml^2} = \frac{n^2\pi^2}{2\tilde{l}^2}\frac{\hbar^2}{\mu_{\text{length}}^2 m} \tag{5.7}$$

where we just factored the length $l = \mu_{\text{length}} \cdot \tilde{l}$. If we define as before $\mu_{\text{energy}} = \frac{\hbar^2}{m\mu_{\text{length}}^2}$ we find

$$\tilde{E}_n = \frac{E_n}{\mu_{\text{energy}}} = \frac{n^2\pi^2}{2\tilde{l}^2}\frac{\hbar^2}{\mu_{\text{length}}^2 m}\frac{\mu_{\text{length}}^2 m}{\hbar^2} = \frac{n^2\pi^2}{2\tilde{l}^2} \tag{5.8}$$

which reproduces an equation of the exact same form, but with $\hbar = m = 1$.
□

Now we look at the time-dependent Schrödinger equation, but inserting $\hat{H} = \mu_{\text{energy}} \cdot \hat{\tilde{H}}$ and $t = \mu_{\text{time}} \cdot \tilde{t}$

$$i\hbar\frac{\partial}{\partial t}\psi = i\frac{\hbar}{\mu_{\text{time}}}\frac{\partial}{\partial \tilde{t}}\psi = \hat{H}\psi = \mu_{\text{energy}} \cdot \hat{\tilde{H}}\psi \Rightarrow \frac{\hbar}{\mu_{\text{time}} \cdot \mu_{\text{energy}}} i\frac{\partial}{\partial \tilde{t}}\psi = \hat{\tilde{H}}\psi \tag{5.9}$$

We have not yet chosen μ_{time} so we are free to require

$$1 = \frac{\hbar}{\mu_{\text{time}} \cdot \mu_{\text{energy}}} = \frac{m \cdot \mu_{\text{length}}^2}{\mu_{\text{time}} \cdot \hbar} \Rightarrow \mu_{\text{time}} = \frac{m \cdot \mu_{\text{length}}^2}{\hbar} \tag{5.10}$$

Upon choosing this we obtain

$$i\frac{\partial}{\partial \tilde{t}}\psi = \hat{\tilde{H}}\psi = \left[-\frac{1}{2}\frac{\partial^2}{\partial \tilde{x}^2} + \tilde{V}\right]\psi \tag{5.11}$$

which is exactly the TDSE, but with $\hbar = m = 1$. To get a fully dimensionless set of equations we can write $\psi = \mu_{\sqrt{1/length}} \cdot \tilde{\psi}$, where the dimensionality trivially divides out on both sides. Now we have produced a new set equations completely identical to the original ones, but have $\hbar = m = 1$. The equations are also dimensionless as they only contain the scaling/simulation values. All that is left in this case is to choose μ_{length}, which again can be conveniently used to simplify equations. For example for the tweezer potential in equation 3.3 we might choose $\mu_{\text{length}} = \sigma$. We could also simply choose something a bit more arbitrary like $\mu_{\text{length}} = 10^{-6}\text{m}$ if that is a good length scale for the problem under consideration.

For convenience we can now slightly modify our notation as to not be so cumbersome to write. We drop the $\sim$ on the simulation values and instead label the SI quantities with subscripts:

$$a_{\text{SI}} = \mu_{[\text{a}]} \cdot a \tag{5.12}$$

Units and Non-dimensionalization

Write each quanitity of interest in the product form $a_{\text{SI}} = \mu_{[\text{a}]} \cdot a$, e.g.

$$L_{\text{SI}} = \mu_{\text{length}} \cdot L, \qquad T_{\text{SI}} = \mu_{\text{time}} \cdot T, \qquad E_{\text{SI}} = \mu_{\text{energy}} \cdot E, \qquad \dots \tag{5.13}$$

where L_{SI}, T_{SI}, E_{SI} are generic SI quantities with dimensions of length, time, and energy, respectively. The $\mu_{[\text{a}]}$'s are then the new units and the a's are the simulation/scaling values.

The equations under consideration can be simplified by inserting the product and make suitable choices for the units $\mu_{[\text{a}]}$. In particular, equations with $\hbar = m = 1$

$$\hat{H} = \left[-\frac{1}{2}\frac{\partial^2}{\partial x^2} + V\right], \qquad \hat{H}\psi = E_n\psi_n, \qquad i\frac{\partial}{\partial t}\psi = \hat{H}\psi \tag{5.14}$$

is obtained if we define

$$\mu_{\text{energy}} = \frac{\hbar^2}{m\mu_{\text{length}}^2}, \qquad \mu_{\text{time}} = \frac{m \cdot \mu_{\text{length}}^2}{\hbar} \tag{5.15}$$

We then just have to define μ_{length}.

For example if $m = m_{Rb}$ and $\mu_{\text{length}} = 10^{-6}\text{m}$ we have $\mu_{\text{energy}} \approx 4.89 \cdot 10^{-13}\text{eV}$, $\mu_{\text{time}} \approx 1.35\text{ms}$.

References

[1] *Introduction to Quantum Mechanics*, David J. Griffith, Pearson Prentice Hall, 2005

[2] *Modern Quantum Mechanics – Revised edition*, J. J. Sakurai, Addison Wesley, 1993

[3] *Quantum and Atom Optics*, Daniel A. Steck, available online at http://steck.us/teaching (revision 0.12.0, 16 May 2017).

[4] *Numerical Optimization* Second Edition, Jorge Nocedal, Stephen J Wright, 2006

[5] *Computational techniques for a quantum control problem with H1-cost,* G von Winckel and A Borzi 2008 Inverse Problems 24 034007

[6] *Time-dependent Quantum Molecular Dynamics,* D. Tannor, V. Kazakov, and V. Orlov, edited by J. Broeckhove and L. Lathouwers (Plenum, New York, 1992), pp. 347–360.