Eva Obeed

Numerical Modelling of Some Quantum Systems

Thesis for the degree of Master in Physics

Trondheim, May 2016

Norwegian University of Science and Technology
Faculty of Natural Sciences and Technology
Department of Physics

Supervisor: Kåre Olaussen

NTNU
Innovation and Creativity

NTNU

Norwegian University of Science and Technology

Thesis for the degree of Master in Physics.

Faculty of Natural Sciences and Technology
Department of Physics

ISBN (printed version)
ISBN (electronic version)
ISSN 1503-8181

Master thesis at NTNU,

Printed by NTNU-trykk

Abstract

Quantum mechanics, like classical mechanics, can be formulated and analyzed in many different ways. Some of these approaches are reviewed in this thesis, to some extent from the viewpoint of how a formulation can be modelled and solved numerically. In this thesis the numerical simulation of two problems is studied.

The first problem is about direct numerical modelling of Heisenberg's matrix mechanics, implemented by replacing an explicit (infinite) matrix representation of position ($\hat{q}$) and momentum ($\mathrm{i}\hat{p}$) operators by $N \times N$ real matrices. A program for explicit generation and numerical analysis of Hamiltonians of the form

$$H = \sum_{j=1}^{D} \left(a_j^{(2)} \hat{p}_j^2 + b_j^{(2)} \hat{q}_j^2 + b_j^{(4)} \hat{q}_j^4 \right) + \sum_{1 \le j < k \le D} b_{j,k}^{(2,2)} \hat{q}_j^2 \, \hat{q}_j^2$$

is presented, and discussed for $D = 1$, 2, 3.

The second problem is numerical modelling of a two-state atom coupled to a collection of photons. We assume an initial Schrödinger "wave function" describing the atom in the excited state, with no photons present. We calculate the time-dependent wave function for this system numerically, and compare the resulting (excited atom) survival probability with the exponential decay rate predicted by perturbation theory and the Fermi Golden Rule.

All simulations are done in `Python`, using numerical routines from the `NumPy` and `SciPy` packages.

Acknowledgement

I would like to gratefully acknowledge various (several) people who have endured with me all these years as I have worked on my thesis.
First, I would like to express my sincere gratitude to *prof.* **Kåre Olaussen**. I'm forever in debt of gratitude to his precious time, support and patience helping me develop my skills on Python. His guidance during my research and writing the thesis, was one of the main reasons, I managed to reach my goal. I couldn't have imagined a better advisor and mentor for my master study.
In addition, I would like to thank my family for their continuous support, by dedicating this thesis to my parents. My father **Nassar Obeed** has always encouraged me to fulfill the dream to be a physicist. He has been my source of inspiration and financial support. Without his support, my studies would have been much harder. Beside my father stood the best mother on earth **Mahasen Hanoush**. I can't find enough words to thank or acknowledge her. She supported me in every possible way with her unconditional love. She believed in me and always helped me to be a better person. "*My dearest mommy, you have been and always will be my source of power. I am who I am today because of you*".
My sincere thanks go to my brother and close friend **Khalil Obeed** for his wise advices that helped me keep my sanity throughout the stressful periods. Also to my lovely sister **Safaa Obeed** and her husband **Ali Shanoon** whom I consider a brother and a wonderful friend, for believing in me.
Very special thanks to the best thing that ever happened to me, to the apple of my heart and soul, to my son **Omar Jarrar**, for understanding when he saw his mommy spending more time studying instead of spending this time with him. He has always been my source of joy under the stressful times through all these years.
Last but not least I would like to express my gratitude to all my friends who have supported and encouraged me during this intensive journey.
Thank you very much, everyone!

Contents

List of Figures

Chapter 1

Introduction

Until the end of the 19^{th} century, classical mechanics based on Newtonian Mechanics and Gravity and Maxwell's equations of Electricity and Magnetism was sufficient to explain all physical phenomena.
Statistical Mechanics was also a well developed mechanism to describe systems with a large number of degrees of freedom. At that time Einstein introduced his Special Relativity theory which could generalized the classical mechanics to include the high velocity systems, where the speed of light C plays a fundamental role.[1]

However, at the late of 19^{th} and early 20^{th} there were many problems remained unexplained. As Black Body Radiation, the Photoelectric effect, basic Atomic Theory, Compton Scattering, and the diffraction of particles. But by Planck's revolution hypothesis of energy quantization $E = h\nu$, where ν is the frequency and h is the minimum quantum actions called Planck's constant, which could explained the full spectrum of the thermal radiation through the blackbody experiment.

Ultimately, working on these problems led to the development of Quantum Mechanics based on the particles -wave duality principle.[2, 3]

Quantum mechanics is one of the most important and successful mathematical model for describing our physical reality, but as Feynman said;

"It is impossible, absolutely impossible to explain it in any classical way"[4]

The fact that QM can not explained in classical way[1] was the motivation for new ways to formulate QM in order to expand our intuition. Quantum Mechanics provided a new understanding of nature different from what

[1] An isolated subsystem of a statistical ensemble with infinitely many classical states may describe a quantum system.

Classical Mechanics provided. While the classical laws of physics are deterministic *(Event A with physical parameter a,b leads to outcome B with physical parameter x,y)*, QM is probabilistic *(Event A leads to outcomes B with probability P1 and outcome C with probability P2 and so on)*.

During the development of the Quantum Mechanics theory, several formulations of Quantum Mechanics emerged side by side with the interpretation of the theory, as physicists need it to facilitates the physical interpretation processes of problems. We have to make this distinction between the formulation and the interpretation of the QM. Where values of physical observables such as energy and momentum no longer considered as values of functions on phase space, but as eigenvalues, as spectral values of linear operators or matrices in Hilbert space [5, 6, 7, 8].

Quantum mechanics may be formulated and analysed in many different ways. As the matrix formulation (Heisenberg) section 2.3, the wavefunction formulation (Schrödinger) section 2.4, Dirac formulation section 2.5, The pilot wave formulation (de Broglie-Bohm) section 2.6, phase space formulation (Wigner) section 2.7, and the path integral formulation (Feynman) section 2.8,... *etc.*

The rest of this thesis is outlined as follows: In Chapter 2, we introduce several formulation to quantum mechanics. In Chapter 3, we introduce spectral theory as an important mathematical tool in our thesis. In Chapter 4, we introduce spectral analysis in Python. In Chapter 5 we take a step further by introducing Harmonic Oscillators step by step starting with classical unforced and undamped harmonic oscillator up to forced and damped oscillator. In Chapter 6 a more detailed descriptions of our numerical simulations is given. Finally, In Chapter 7, we summarize our work.

Chapter 2

The many formulations of (Quantum) Mechanics

Quantum Mechanics is the foundation of most active areas of contemporary physics, such as atomic and molecular physics, solid state physics, and particle physics. I.e., where the aim is to describe objects which are very small, and phenomena which occur very fast, compared to everyday experience. An accurate dynamical description of such cases cannot be achieved by the classical mechanics of Newton, or its Lagrangian and Hamiltonian reformulations.

Since there is no obvious line of separation between what should be considered "large" and "small", or "fast" and "slow", the same theory ought (in principle) to be able to describe all kinds of phenomena, only with possible simplifications in various limiting cases. Therefore, classical mechanics should be expected to be an approximation of quantum mechanics in some limits, which must include the description of ordinary macroscopic objects. It may therefore not be surprising that the new descriptions, which have be realized through many (seemingly) different formulations of quantum mechanics, to a large degree are based on the old mechanics section 2.2 and its extention to classical statistical mechanics. Before we discuss various quantum formulations it is natural of review some parts of classical mechanics.

2.1 Review of classical mechanics

2.1.1 Newton's second law of mechanics

One may say that the most basic equation of standard mechanics is Newton's second law of motion,

$$\boldsymbol{F} = m\boldsymbol{a}, \tag{2.1}$$

where m is the mass of a point particle with position $\boldsymbol{r}(t)$, and $\boldsymbol{a} = \ddot{\boldsymbol{r}}$ is its acceleration. The concept of a *force* $\boldsymbol{F}$ is therefore the most essential part of the Newtonian formulation. As it stands, equation (2.1) has no predictive content before an independent expression for the force known. A common class are *conservative forces*, which can be expressed as the gradient of a potential V,

$$\boldsymbol{F} = \boldsymbol{F}(\boldsymbol{r}) = -\boldsymbol{\nabla} V(\boldsymbol{r}). \tag{2.2}$$

One important property of such forces is that the *energy*

$$E = \frac{1}{2} m\boldsymbol{v}^2 + V(\boldsymbol{r}), \tag{2.3}$$

remain unchanged during the motion, where $\boldsymbol{v} = \dot{\boldsymbol{r}}$ is the particle velocity.

2.1.2 Principle of stationary action

Some of the information in this subject taken from the Wikipedia article [9].

For conservative forces, and some generalizations, there are alternative formulations of a dynamical principle. One formulation starts by assigning an *action*,

$$S\{\boldsymbol{q}\} = \int_{t_i}^{t_f} dt\; L(\boldsymbol{q}, \dot{\boldsymbol{q}}), \tag{2.4}$$

to every continuous path $\boldsymbol{q} = \boldsymbol{q}(t)$ from $\boldsymbol{q}_i \equiv \boldsymbol{q}(t_i)$ to $\boldsymbol{q}_f \equiv \boldsymbol{q}(t_f)$. I.e., the action is a *functional* of the path, constructed as the time-integral of a *Lagrange function* $L(\boldsymbol{q}, \dot{\boldsymbol{q}}) \equiv L(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))$.

Hamilton's principle, or the *principle of stationary action*[1], says that of all possible paths $\boldsymbol{q}$ the classical solution $\boldsymbol{q}_c$ is one which makes the action stationary. I.e., $\boldsymbol{q}_c$ is selected such that

$$\delta S \equiv \frac{d}{d\epsilon} S\{\boldsymbol{q}_c + \epsilon\boldsymbol{\delta q}\}\bigg|_{\epsilon=0} = \int_{t_i}^{t_f} dt \left(\frac{\partial L}{\partial q^a}\delta q^a + \frac{\partial L}{\partial \dot{q}^a}\delta \dot{q}^a \right) = 0, \tag{2.5a}$$

for all possible deviations $\boldsymbol{\delta q} = \boldsymbol{\delta q}(t)$ satisfying $\boldsymbol{\delta q}(t_i) = \boldsymbol{\delta q}(t_f) = 0$. In the integral above, (i) the vectors $\boldsymbol{q}$ and $\boldsymbol{\delta q}$ are assumed to have components

[1] Generally, the action simply stationary, neither minimum nor maximum.

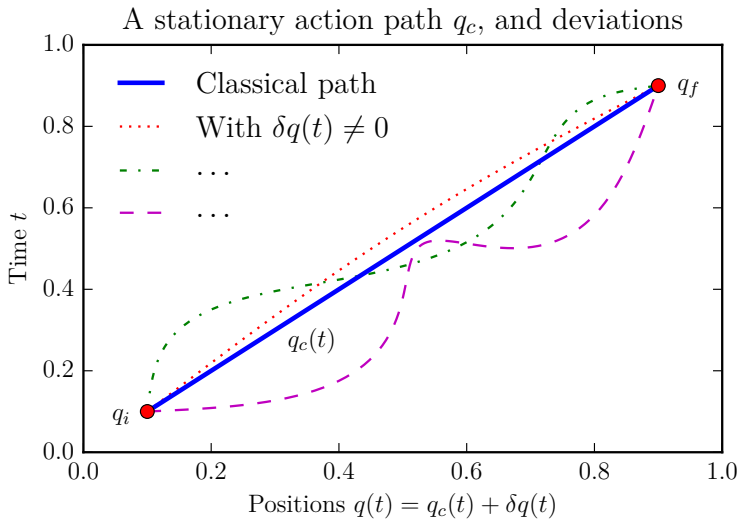


Figure 2.1: Of all possible paths starting at position q_i at time t_i, and ending at position q_f at time t_f, classical dynamics selects those which makes the action stationary. Here indicated by the thick fulldrawn (blue) line.

q^a and δq^a, (ii) the integrand should be evaluated on the classical path, $q^a(t) = q_c^a(t)$, and (iii) there is an implicit sum over repeated indices a (the Einstein summation convention). The physical content of equation (2.5a) is illustrated in figure 2.1 for the case of motion in empty space. In such cases the classical particle must move on a straight line with constant velocity — in agreement with Newton's first law of mechanics.

After integrating the 2nd term in the integrand of equation (2.5a) by parts, one obtains

$$\delta S = \int_{t_i}^{t_f} dt \, \left[\frac{\partial L}{\partial q^a} - \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}^a} \right) \right] \delta q^a = 0. \tag{2.5b}$$

It has been used that the endpoint contribution, $(\partial L/\partial \dot{q}^a)\, \delta q^a|_{t_i}^{t_f}$, vanishes because $\delta q^a = 0$ at the endpoints. Since equation (2.5b) must hold for all possible $\boldsymbol{\delta q}$, this leads to the *Euler-Lagrange equations*,

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}^a} \right) = \frac{\partial L}{\partial q^a} \text{ for all } a. \tag{2.6}$$

A connection the Newtonian mechanics of subsection 2.1.1 is made by choos-

ing a Lagrange function of the form

$$L(\boldsymbol{q}, \dot{\boldsymbol{q}}) = T(\dot{\boldsymbol{q}}) - V(\boldsymbol{q}). \tag{2.7}$$

For a single point particle, $\boldsymbol{q}$ can be choosen to be the position vector $\boldsymbol{r}$ of that particle, with

$$T(\dot{\boldsymbol{r}}) = \frac{1}{2} m \dot{\boldsymbol{r}}^2 \tag{2.8}$$

its (non-relativistic) kinetic energy, and $V(\boldsymbol{r})$ its potential energy. Equations (2.6) become

$$\frac{d}{dt} m \dot{r}^a = m \ddot{r}^a = -\frac{\partial V}{\partial r^a},$$

which is the component form of Newton's second law with a conservative force $\boldsymbol{F} = -\boldsymbol{\nabla} V$. This can be extended a collection of N point particles with masses m_i, by choosing $\boldsymbol{q}$ to be a vector with $3N$ components,

$$\boldsymbol{q} = (\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N), \qquad \text{and} \qquad T(\dot{\boldsymbol{q}}) = \sum_{i=1}^{N} m_i \dot{\boldsymbol{r}}^2.$$

The potential is often a sum of pair potentials,

$$V(\boldsymbol{q}) = \frac{1}{2} \sum_{i \neq j} V_{ij}(|\boldsymbol{r}_i - \boldsymbol{r}_j|).$$

As examples, for Newtonian forces of gravity, $V_{ij}^{(G)} = -m_i m_j G_N / r$, where $G_N \approx 6.674 \times 10^{-11}\ \mathrm{m^3 s^{-3} kg^{-1}}$ is the gravitational constant. For Coulomb forces, $V_{ij}^{(C)} = Q_i Q_j / (4\pi\varepsilon_0 r)$, where $\varepsilon_0 = 8.854 \times 10^{-12}\ \mathrm{m^{-3} kg^{-1} s^4 A^2}$ is the permittivity of free space, and Q_i is the charge of particle i measured in Coulomb (C $\equiv$ As).

For particles in a magnetic field $\boldsymbol{B} = \nabla \times \boldsymbol{A}(r_i)$, the kinetic energy become

$$T(\dot{\boldsymbol{q}}) \to T(\dot{\boldsymbol{q}}) + e \dot{\boldsymbol{q}}.A$$

where,

$$e \dot{\boldsymbol{q}}.A \equiv \sum_{n=1}^{N} Q_i . \dot{\boldsymbol{r}}_{\boldsymbol{i}} . \boldsymbol{A}(r_i)$$

In relativistic mechanics,

$$\frac{1}{2} m \dot{\boldsymbol{r}}^2 = -mc \sqrt{1 - \left(\frac{\dot{\boldsymbol{r}}}{c}\right)^2}.$$

And[2]

$$T(\dot{\boldsymbol{q}}) = \frac{1}{2}\sum_i m_i \dot{\boldsymbol{r}}_i^2 \to -\sum_i m_i c^2 \sqrt{1-\left(\frac{\dot{\boldsymbol{r}}_i}{c}\right)^2}.$$

2.1.3 Hamiltonian mechanics

Some of the information in this subject taken from the Wikipedia article [10].

The Euler-Lagrange equations (2.6) can be written in the form

$$\frac{dp_a}{dt} = \frac{\partial L}{\partial q^a}, \tag{2.9}$$

where

$$p_a \equiv (\partial L/\partial \dot{q}^a) \tag{2.10}$$

is the *canonically conjugate momentum* of the coordinate q^a. One may say that this is the general version of Newton's second law. Note that p_a will remain constant if L does not depend explicitly on q^a, $\partial L/\partial q^a = 0$. This is one reason to eliminate velocities $\dot{q}^a$ in favor of conjugate momenta p_a. At the same time the Lagrange function L is replaced by the Hamilton function (or *Hamiltonian*),

$$H(\boldsymbol{q},\boldsymbol{p}) = p_a\dot{q}^a - L(\boldsymbol{q},\dot{\boldsymbol{q}}), \tag{2.11}$$

where $\dot{\boldsymbol{q}}$ should be interpreted to be a function of $\boldsymbol{p}$, and possibly also $\boldsymbol{q}$. I.e., $\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}(\boldsymbol{q},\boldsymbol{p})$. The transformation from a Lagrangian to Hamiltonian formalism,

$$(\boldsymbol{q},\dot{\boldsymbol{q}}) \to (\boldsymbol{q},\boldsymbol{p}) \quad \text{with} \quad L(\boldsymbol{q},\dot{\boldsymbol{q}}) \to H(\boldsymbol{q},\boldsymbol{p}),$$

is an example of a *Legendre transformation*. The equations of motion can now be written in the form of Hamilton's equations,

$$\begin{pmatrix}\dot{q}^a\\ \dot{p}_a\end{pmatrix} = \begin{pmatrix}\partial H/\partial p_a\\ -\partial H/\partial q^a\end{pmatrix}. \tag{2.12}$$

By introducing the *Poisson bracket* $\{\cdot,\cdot\}$ between two functions A, B of $\boldsymbol{q}$ and $\boldsymbol{p}$,

$$\{A,B\} \equiv \sum_b \left[\left(\frac{\partial A}{\partial q^b}\right)\left(\frac{\partial B}{\partial p_b}\right) - \left(\frac{\partial B}{\partial q^b}\right)\left(\frac{\partial A}{\partial p_b}\right)\right]. \tag{2.13}$$

[2]
$$\sqrt{1-\left(\frac{\dot{\boldsymbol{r}}_i}{c}\right)^2} = \left(1-\frac{1}{2}\frac{\dot{\boldsymbol{r}}_i^2}{c^2}+\dots\right)$$

This implies in particular that

$$\{q^a, p_b\} = \delta^a_b, \quad \{q^a, q^b\} = \{p_a, p_b\} = 0. \tag{2.14}$$

Hamilton's equations can be written

$$\dot{z}^a = \{z^a, H\}\,, \tag{2.15}$$

where $\boldsymbol{z} = \boldsymbol{q}$ or $\boldsymbol{p}$. More generally in Hamiltonian mechanics, the time dependence of any "physical quantity" (observable) $\mathcal{O} \equiv \mathcal{O}(\boldsymbol{q}, \boldsymbol{p}; t)$ is given by

$$\frac{d\mathcal{O}}{dt} = \{\mathcal{O}, H\} + \frac{\partial \mathcal{O}}{\partial t}. \tag{2.16}$$

I.e., the requirement for an observable (which does not depend explicitly on time) to remain constant is that it must have a vanishing Poisson bracket with the Hamiltonian.

All descriptions above have assumed that the state of the mechanical system is exactly known at each time t, as specified by the quantities $\boldsymbol{q}(t)$ and $\boldsymbol{p}(t)$. This is a mathematical idealization. A more realistic description is to define a *probability density* $\rho \equiv \rho(\boldsymbol{q}, \boldsymbol{p}; t)$, so the probability for the system to be in a volume element $\mathrm{d}\boldsymbol{q}\,\mathrm{d}\boldsymbol{p}$ around the point $(\boldsymbol{q}, \boldsymbol{p})$ is $\rho\,\mathrm{d}\boldsymbol{q}\,\mathrm{d}\boldsymbol{p}$. This means that the value of each observable $\mathcal{O}$ is no longer exactly known, but must be specified statistically by quantities like its mean value and variance,

$$\bar{\mathcal{O}} = \int \mathcal{O}(\boldsymbol{q}, \boldsymbol{p})\, \rho(\boldsymbol{q}, \boldsymbol{p}; t)\, \mathrm{d}\boldsymbol{q}\, \mathrm{d}\boldsymbol{p}, \tag{2.17a}$$

$$\mathrm{Var}\,(\mathcal{O}) = \int \left[\mathcal{O}(\boldsymbol{q}, \boldsymbol{p}) - \bar{\mathcal{O}}\right]^2 \rho(\boldsymbol{q}, \boldsymbol{p}; t)\, \mathrm{d}\boldsymbol{q}\, \mathrm{d}\boldsymbol{p}. \tag{2.17b}$$

The limit of an exactly known state is obtained by the probability density

$$\rho(\boldsymbol{q}, \boldsymbol{p}; t) = \prod_a \delta(q^a - q^a(t))\, \delta(p^a - p^a(t)). \tag{2.18}$$

The Hamilton equations for the probability density becomes

$$\frac{d\rho}{dt} = -\{\rho, H\}\,. \tag{2.19}$$

Note that there is a difference in sign between eqs.(2.16) and (2.19)! There is an alternative to (2.17) for finding the time dependence of expectation values: One may instead assume the probability density to be time independent, $\rho = \rho(\boldsymbol{q}, \boldsymbol{p}; 0)$, and use observables which vary in time according to equation (2.16), $\mathcal{O} = \mathcal{O}(\boldsymbol{q}, \boldsymbol{p}, t)$.

2.2 Old quantum mechanics (1913)

Some of the information in this subject taken from the Wikipedia article [11].

One may say the quantum mechanical description of matter started with the Bohr model in 1913 [12, 13, 14]. Earlier at the start of 20th century, experiments of scattering alpha particles on matter [15, 16] had shown that large deflections could occationally occur. This led Rutherford to propose a model the atom structure as a "miniature solar system", with electrons orbiting the central nucleus [17]. But this model should imply that atoms

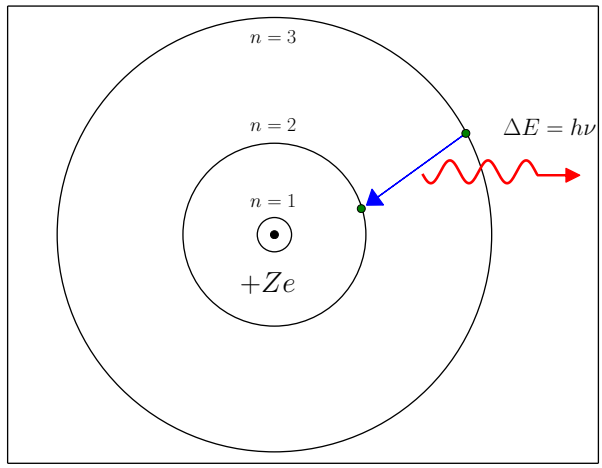


Figure 2.2: The Rutherford–Bohr model of the hydrogen atom ($Z = 1$), where an electron jump between orbits emit or absorb amount of electromagnetic energy ($\Delta E = h\nu$).

were unstable, since the orbiting electrons would emit electromagnetic radiation and lose energy while spiralling into the nucleus within a fraction of a second. In other words, according to classical theory of mechanics and electrodynamics, Rutherford's atomic solar systems should collapse, releasing an enormous amount of energy in the process (compared to observed atomic transition energies).

This led Bohr to propose an amended model of the atom, according to which it can be stable in a certain set of orbits, with a corresponding discrete set of energies E_n. External perturbations might then cause the

electrons to move from one stable orbit to another, and in the process emit or absorb a discrete amount of electromagnetic energy in the form of one or more electromagnetic quanta — photons. For the transition from an orbit with energy E_i to an orbit with energy $E_f < E_i$, with the emission of a single photon, the photon energy would be $E_i - E_f$. Hence, according to the Einstein 1905 hypothesis [18], it should have a frequency ν such that

$$h\nu \equiv \hbar\omega = (E_i - E_f), \tag{2.20}$$

where $h = 2\pi\hbar = 6.626 \times 10^{-34}\ \mathrm{kg\,m^2\,s^{-1}}$ is the Planck constant.

What should be the condition for a stable orbit? Following Bohr [12], consider a electron of mass m_e and charge $-e$ in a circular orbit of radius r around a very much heavier nucleus of charge Ze. For a nucleus of infinite mass, Newton's second law becomes the condition

$$m_e v^2/r = Ze^2/(4\pi\varepsilon_0 r^2). \tag{2.21}$$

This implies that $V = -Ze^2/(4\pi\varepsilon_0 r) = -m_e v^2 = -2T$. By assuming that the electron angular momentum must be an integer multiple of $\hbar$,

$$L = m_e v r = n\hbar, \tag{2.22}$$

one obtains a condition for the allowed types of circular orbits. By combining equations (2.21) and (2.22) one finds for the energy, velocity and radius:

$$E_n = -\frac{(Z\alpha)^2}{2n^2}\, m_e c^2, \tag{2.23a}$$

$$v_n = \frac{Z\alpha}{|n|}\, c, \tag{2.23b}$$

$$r_n = \frac{n^2}{Z\alpha}\left(\frac{\hbar}{m_e c}\right). \tag{2.23c}$$

Here $\alpha \equiv e^2/(4\pi\varepsilon_0\hbar c) \approx 1/137.036$ is the dimensionless Sommerfeld constant, and $\hbar/(m_e c) \approx 3.862 \times 10^{-13}\,\mathrm{m}$ is the electron Compton wavelength. For $Z = 1$ and $n = 1$ (ground state of hydrogen atom) one obtains $|E_1| \approx 13.606$ eV (Rydberg energy), $r_1 = 0.539 \times 10^{-10}$ m (Bohr radius), and $v_1 \approx 2\,188$ km/s.

This gives for the possible transition energies in equation (2.20),

$$E_i - E_f = |E_1|\left(n_f^{-2} - n_i^{-2}\right), \tag{2.24}$$

in good agreement with the Balmer ($n_f = 2$, $n_i = 3, 4, \ldots$) and Paschen ($n_f = 3$, $n_i = 4, 5, \ldots$) series [19, 20], which had been observed in stellar

hydrogen spectra before 1913. By replacing the electron mass m_e with the reduced mass of a two-body system, $m_e \to (1 + m_e/m_N)^{-1} m_e$, where m_N is the mass of the nucleus, a further improvement is obtained.

Although the derivation above gives a phenomenologically very successful description of atomic physics, at least for the hydrogen atom, it is not quite correct. The most obvious flaw is that it does *not* predict a stable atom, since an electron with angular momentum $L = 0$ ($n = 0$) would have energy $E_0 = -\infty$. Or at least very negative, if one assumes that the nucleus to have some kind of short range repulsion which prevents a complete collapse. In reality, the (lowest) energy of an electron with angular momentum $L = n\hbar$ is $E_{n+1} = E_1/(n + 1)^2$. A more general condition for stable orbits was

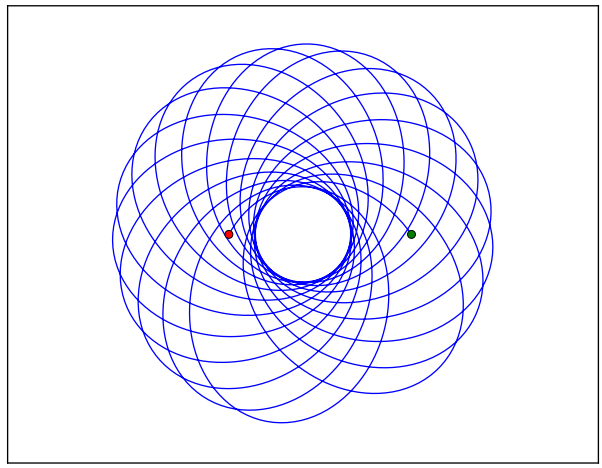


Figure 2.3: *The motion of the electron for the hydrogen atom (starting from the green point) according to classical relativistic mechanics [21].*

proposed independently by Wilson [22] and Sommerfeld [23]. According to this a periodic, stable orbit should satisfy the relation

$$\oint p_k dq_k = 2\pi n_k \hbar, \tag{2.25}$$

for each pair (q_k, p_k) of (generalized) coordinate and conjugate momenta. For example, for a harmonic oscillator defined by the Hamiltonian $H =$

$\frac{1}{2m}p^2 + \frac{1}{2}m\omega^2 q^2$, the condition for an orbit with energy E becomes

$$2\int_{-q_{\max}}^{q_{\max}} \sqrt{2mE - m^2\omega^2 q^2}\, dq = 2\pi n\hbar, \tag{2.26}$$

where $m^2\omega^2 q_{\max}^2 = 2mE$. The value of the integral (including the factor 2 in front) is $\pi m\omega q_{\max}^2 = 2\pi E/\omega$. Hence, equation (2.25) predicts that $E_n = n\hbar\omega$. I.e., that the energy of harmonic oscillations must vary in integer units of $\hbar\omega = h\nu$. This is correct, although the further development of quantum mechanics leads to an overall shift of the above prediction,

$$E_n = (n + \tfrac{1}{2})\hbar\omega, \qquad n = 0, 1, \ldots. \tag{2.27}$$

This result is in agreement with the Einstein hypothesis that the energy of light comes in integer units of $h\nu$. In fact, the mechanical description of the electromagnetic field is mathematically equal to an infinite collection of harmonic oscillators; hence light is just a particular case of the general description.

Consider now the previous electron-nucleus system with purely radial motion, $L = 0$, described by the Hamiltonian $H = \frac{1}{2m_e}p_r^2 - Ze^2/(4\pi\varepsilon_0 r)$. The condition for a bound orbit with energy $E < 0$ becomes

$$2\int_0^{r_{\max}} \sqrt{2m\left[Ze^2/(4\pi\varepsilon_0 r) - |E|\right]}\, dr = 2\pi n_r\hbar, \tag{2.28}$$

with $Ze^2/(4\pi\varepsilon_0 r_{\max}) = |E|$. The value of the integral (including the factor 2 in front) is $(Ze^2/4\pi\varepsilon_0)\sqrt{2m/|E|}\,\pi$. It follows that equation (2.28) also leads to the energy levels (2.23a),

$$E_{n_r} = -\frac{(Z\alpha)^2}{2n_r^2} m_e c^2. \tag{2.29}$$

By insisting that n_r must be a positive integer, $n_r = 1, 2, \ldots$, only stable orbits are allowed[3]. However, the reason for this requirement — like the conditions (2.25) themselves — remains unclear. In the further development of quantum mechanics the conditions (2.25) can derived as a semi-classical approximation, usually good only for large values of n.

2.3 Heisenberg Matrix Formulation (1925)

The Heisenberg reformulation [24] of quantum mechanics was motivated by the problem of understanding transitions between the stable orbits of old

[3] Actually, it appears that Sommerfeld rejected all $L = 0$ orbits as unphysical.

quantum mechanics. According to classical theory of radiation, the emitted light should have a frequency corresponding to the oscillation frequency of the radiating electron. This does not quite match with equation (2.20).

For a transition from orbit m to n, Heisenberg postulated a time dependence

$$q_{mn}(t) = \mathrm{e}^{\mathrm{i}(E_m - E_n)t/\hbar}\, q_{mn}(0), \tag{2.30}$$

in order to obtain agreement with equation (2.20). Further explorations of this idea led to a *quantum jump* picture of atomic transitions which cannot be described in terms of classical electron orbits. In fact, the "positions" $q_{mn}(t)$ had to be non-commuting objects. This was soon developed further into a full theory of matrix mechanics [25, 26, 27], where each classical coordinate q and conjugate momentum p is replaced by a matrix (infinite-dimensional), $\hat{q}$ resp. $\hat{p}$. One explicit example is the representation

$$\hat{q} = \sqrt{\frac{\hbar}{2\bar{m}\bar{\omega}}} \begin{pmatrix} 0 & \sqrt{1} & 0 & 0 & \cdots \\ \sqrt{1} & 0 & \sqrt{2} & 0 & \cdots \\ 0 & \sqrt{2} & 0 & \sqrt{3} & \cdots \\ 0 & 0 & \sqrt{3} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \tag{2.31a}$$

$$\hat{p} = -\mathrm{i}\sqrt{\frac{\bar{m}\bar{\omega}\hbar}{2}} \begin{pmatrix} 0 & \sqrt{1} & 0 & 0 & \cdots \\ -\sqrt{1} & 0 & \sqrt{2} & 0 & \cdots \\ 0 & -\sqrt{2} & 0 & \sqrt{3} & \cdots \\ 0 & 0 & -\sqrt{3} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}. \tag{2.31b}$$

These matrices satisfy the commutation relation[4]

$$[\hat{q}, \hat{p}] \equiv (\hat{q}\,\hat{p} - \hat{p}\,\hat{q}) = \mathrm{i}\hbar, \tag{2.33}$$

which is formally very similar to the Poisson brackets (2.14) between coordinates and conjugate momenta. In fact, most concepts and equations of Hamiltonian mechanics, as reviewed in subsection 2.1.3, can formally be

[4] For later use is convenient to introduce matrices (annihilation and creation operators) a and $a^\dagger$ such that

$$\hat{q} = \sqrt{\frac{\hbar}{2\bar{m}\bar{\omega}}} \left(a + a^\dagger\right), \tag{2.32a}$$

$$\hat{p} = -\mathrm{i}\sqrt{\frac{\bar{m}\bar{\omega}\hbar}{2}} \left(a - a^\dagger\right) \tag{2.32b}$$

Here $a^\dagger$ is the hermittian conjugate of a, as the notation suggests, and the commutator $\left[a, a^\dagger\right] = 1$. One further finds that the product $a^\dagger a$ is diagonal with eigenvalues $0, 1, \ldots$.

carried over to matrix mechanics — only with a different definition of the Poisson bracket $\{\cdot,\cdot\}$. In matrix mechanics the classical definition (2.13) is replaced by the commutator

$$\left\{\hat{A},\hat{B}\right\} = \frac{1}{\mathrm{i}\hbar}\left[\hat{A},\hat{B}\right] \equiv \frac{1}{\mathrm{i}\hbar}\left(\hat{A}\hat{B} - \hat{B}\hat{A}\right). \tag{2.34}$$

With this definition, the most basic mathematical properties of the Poisson bracket (like linearity, antisymmetry, the product rule, and the Jacobi identity) remain unchanged:

$$\{A, bB + cC\} = b\{A,B\} + c\{A,C\}, \quad \text{(with constants } b,c\text{)}, \tag{2.35a}$$
$$\{A,B\} + \{B,A\} = 0, \tag{2.35b}$$
$$\{A, BC\} = \{A,B\}C + B\{A,C\}, \tag{2.35c}$$
$$\{A,\{B,C\}\} + \{C,\{A,B\}\} + \{B,\{C,A\}\} = 0, \tag{2.35d}$$

where we have dropped the $\hat{}$ which indicate matrix quantities (which we shall continue to do in the following). One difference between classical and matrix mechanics is that one must be careful with ordering of terms, as in equation (2.35c), since the quantities may no longer commute. Consider as example equation (2.15) with Hamiltonian $\frac{1}{2m}p^2 + \frac{1}{2}m\omega^2 q^2$. One finds

$$\begin{aligned}\dot{q} &= \{q,H\} = \tfrac{1}{2m}\{q,p^2\} = \tfrac{1}{2m}\left(p\{q,p\} + \{q,p\}p\right) = p/m,\\ \dot{p} &= \{p,H\} = \tfrac{1}{2}m\omega^2\{p,q^2\} = \tfrac{1}{2}m\omega^2\left(q\{p,q\} + \{p,q\}q\right) = -m\omega^2 q,\end{aligned} \tag{2.36}$$

where only general algebraic properties of the quantities involved have been used. Specifically, linearity (2.35a), the basic Poisson brackets (2.14), the product rule (2.35c), and the antisymmetry property (2.35b). By insertion one can verify that

$$q(t) = \cos\omega t\, q_0 + (m\omega)^{-1}\sin\omega t\, p_0, \tag{2.37a}$$
$$p(t) = -(m\omega)\sin\omega t\, q_0 + \cos\omega t\, p_0, \tag{2.37b}$$

is a solution to (2.36), with initial conditions (q_0,p_0) at $t=0$. For instance, one may choose $(q_0,p_0) = (\hat{q},\hat{p})$ as defined in equation (2.31), which gives

$$q_{n+1,n}(t) = \left[\cos\omega t + \mathrm{i}\frac{\bar{m}\bar{\omega}}{m\omega}\sin\omega t\right]\sqrt{\frac{\hbar(n+1)}{2\bar{m}\bar{\omega}}}, \tag{2.38a}$$
$$p_{n+1,n}(t) = \mathrm{i}\left[\cos\omega t + \mathrm{i}\frac{m\omega}{\bar{m}\bar{\omega}}\sin\omega t\right]\sqrt{\frac{\bar{m}\bar{\omega}\hbar(n+1)}{2}} \tag{2.38b}$$

and $q_{n,n+1} = q^*_{n+1,n}$, $p_{n,n+1} = p^*_{n+1,n}$. All other matrix elements are zero. Equations (2.38) agree with Heisenbergs postulate (2.30) when $\bar{m}\bar{\omega} = m\omega$.

For the general quantum case equation (2.16) can be written as

$$\frac{d}{dt}\mathcal{O} = \frac{\mathrm{i}}{\hbar}\,[H, \mathcal{O}] + \frac{\partial \mathcal{O}}{\partial t}. \tag{2.39}$$

Assuming that $\mathcal{O}$ and H have no explicit time dependence, equation (2.39) has solution

$$\mathcal{O}(t) = \mathrm{e}^{\mathrm{i}Ht/\hbar}\,\mathcal{O}(0)\,\mathrm{e}^{-\mathrm{i}Ht/\hbar}. \tag{2.40a}$$

Expressed in a basis where H is diagonal, $\left(\mathrm{e}^{\mathrm{i}Ht/\hbar}\right)_{mn} = \mathrm{e}^{\mathrm{i}E_m t}\delta_{mn}$, this becomes

$$\mathcal{O}_{mn}(t) = \mathcal{O}_{mn}(0)\,\mathrm{e}^{\mathrm{i}(E_m - E_n)t/\hbar}, \tag{2.40b}$$

in general agreement with (2.30).

Hence, in the matrix formulation, the possible energies of a system must be identified with the eigenvalues of the Hamiltonian H, and the stable orbits of Bohr-Sommerfeld quantization are (in some sense) replaced by the corresponding eigenvectors of H. More generally, the best possible description of a system is in terms of a complex *state vector* ψ (with components ψ_k with respect to some basis), normalized to unity: $\sum_k |\psi_k|^2 = 1$. The possible values of a quantized physical quantity (an *observable*) $\mathcal{O}$ are its eigenvalues $\mathcal{O}_n$. Assume the corresponding (complete, orthonormal set of) eigenvectors to be $\chi_{\mathcal{O}}^{(n)}$, i.e.

$$\mathcal{O}\chi_{\mathcal{O}}^{(n)} = \mathcal{O}_n\,\chi_{\mathcal{O}}^{(n)},$$

and expand ψ on this basis, $\psi = \sum_n \psi_n\,\chi_{\mathcal{O}}^{(n)}$. Then, according to the *Born interpretation* [28], the probability of observing the value $\mathcal{O}_n$ in the state ψ is $p_n = |\psi_n|^2$. For a system to have exact values for some observables $\mathcal{O}^{(1)}, \mathcal{O}^{(2)}, \ldots$, it must be described by a state vector ψ which is a simultaneous eigenvector of all these observables.

With matrix mechanics, systems which can be treated with the Sommerfeld quantization rule (2.25) — like the harmonic oscillator, particle angular momentum, and the hydrogen atom — can be solved in an unambiguous way. In particular, Pauli calculated the complete non-relativistic spectrum of the hydrogen atom by matrix methods [29]. The general problem reduces to the problem of finding eigenvalues of the Hamiltonian H, or other observables.

Matrix mechanics differs from classical mechanics in puzzling ways. Assume we have found a vector $\chi^{(n)}$ which describes a particle with exact position q_n and exact momentum p_n, such that

$$q\,\chi^{(n)} = q_n\,\chi^{(n)}, \qquad p\,\chi^{(n)} = p_n\,\chi^{(n)}.$$

Then we must have, since q_n and p_n are ordinary numbers,

$$q\,p\,\chi^{(n)} = q\,p_n\,\chi^{(n)} = p_n q\,\chi^{(n)} = p_n q_n\,\chi^{(n)},$$
$$p\,q\,\chi^{(n)} = p\,q_n\,\chi^{(n)} = q_n p\,\chi^{(n)} = q_n p_n\,\chi^{(n)}.$$

By subtraction it follows that $[q,p]\chi^{(n)} = (q\,p - p\,q)\,\chi^{(n)} = 0$. But this is not consistent with equation (2.33), which implies that $[q,p]\,\chi^{(n)} = \mathrm{i}\hbar\chi^{(n)}$. Hence, a quantum system cannot have exact position and (conjugate) momentum at the same time. This is very different from the circular or elliptical orbits of the Bohr-Sommerfeld description.

How can one recover a classical description, which is known to work very well for mechanical systems of everyday experience? Consider the Hamiltonian

$$H = \frac{\boldsymbol{p}^2}{2m} + V(\boldsymbol{r}), \tag{2.41}$$

where the (matrix) components of each vector $\boldsymbol{p}$ and $\boldsymbol{r}$ commute among themselves, but not with each other:

$$[p_j, p_k] = 0, \quad [r_j, r_k] = 0, \quad [p_j, r_k] = -\mathrm{i}\hbar\delta_{jk}. \tag{2.42}$$

The Heisenberg equations of motion become[5]

$$\begin{aligned}\dot{r}_j &= \frac{i}{\hbar}\,[H, r_j] = \frac{i}{2m\hbar}\,[p_k p_k, r_j] \\ &= \frac{i}{2m\hbar}\,(p_k\,[p_k, r_j] + [p_k, r_j]\,p_k) = \frac{p_j}{m}, \qquad (2.43a)\\ \dot{p}_j &= \frac{i}{\hbar}\,[H, p_j] = \frac{i}{\hbar}\,[V(\boldsymbol{r}), p_j] = -\frac{\partial}{\partial r_j}V(\boldsymbol{r}). \qquad (2.43b)\end{aligned}$$

[5] To derive of the last equality of equation (2.43b), with use of the relations (2.42) only, one can first show that

$$\boldsymbol{r}(t) \equiv \mathrm{e}^{\mathrm{i}t\boldsymbol{u}\cdot\boldsymbol{p}/\hbar}\,\boldsymbol{r}\,\mathrm{e}^{-\mathrm{i}t\boldsymbol{u}\cdot\boldsymbol{p}/\hbar} = \boldsymbol{r} + t\boldsymbol{u},$$

because $\boldsymbol{r}(t)$ satisfies the differential equation

$$\frac{d}{dt}\boldsymbol{r}(t) = \mathrm{e}^{\mathrm{i}t\boldsymbol{u}\cdot\boldsymbol{p}/\hbar}\,\frac{\mathrm{i}}{\hbar}\,[\boldsymbol{u}\cdot\boldsymbol{p}, \boldsymbol{r}]\,\mathrm{e}^{-\mathrm{i}t\boldsymbol{u}\cdot\boldsymbol{p}/\hbar} = \boldsymbol{u},$$

with initial condition $\boldsymbol{r}(0) = \boldsymbol{r}$. From this it follows that

$$\mathrm{e}^{\mathrm{i}t\boldsymbol{u}\cdot\boldsymbol{p}/\hbar}\,V(\boldsymbol{r})\,\mathrm{e}^{-\mathrm{i}t\boldsymbol{u}\cdot\boldsymbol{p}/\hbar} = V(\boldsymbol{r} + t\boldsymbol{u}).$$

By differentiating with respect to t, and setting $t = 0$, one obtains

$$\frac{i}{\hbar}\,[\boldsymbol{u}\cdot\boldsymbol{p}, V(\boldsymbol{r})] = \boldsymbol{u}\cdot\boldsymbol{\nabla}V(\boldsymbol{r}).$$

Equation (2.43b) for $j = 1, 2, 3$ is now obtained by choosing $\boldsymbol{u}$ to respectively $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$.

For a given state vector ψ_n, we define the mean value of observables $\mathcal{O}$ as

$$\langle \mathcal{O} \rangle = \sum_{mn} \psi_m^* \mathcal{O}_{mn} \psi_n. \tag{2.44}$$

Then, by taking the mean value of the matrix equations (2.43), we obtain

$$\langle \dot{\boldsymbol{r}} \rangle = \langle \boldsymbol{p}/m \rangle, \quad \langle \dot{\boldsymbol{p}} \rangle = -\langle \boldsymbol{\nabla} V(\boldsymbol{r}) \rangle,$$

or

$$m\langle \ddot{\boldsymbol{r}} \rangle = -\langle \boldsymbol{\nabla} V(\boldsymbol{r}) \rangle. \tag{2.45}$$

Newton's second law is reproduced for the average position when one can make the approximation $\langle \boldsymbol{\nabla} V(\boldsymbol{r}) \rangle \approx \boldsymbol{\nabla} V(\langle \boldsymbol{r} \rangle)$.

2.3.1 Explicit matrix representations

Choose units in (2.31) such that $\bar{m}\bar{\omega} = 1$ and $\hbar = 1$, and define matrices

$$Q^{(2e)}_{m,n} \equiv \left(\hat{q}^2\right)_{2m,2n}, \tag{2.46a}$$
$$Q^{(2o)}_{m,n} \equiv \left(\hat{q}^2\right)_{2m+1,2n+1}, \tag{2.46b}$$
$$P^{(2e)}_{m,n} \equiv \left(\hat{p}^2\right)_{2m,2n}, \tag{2.46c}$$
$$P^{(2o)}_{m,n} \equiv \left(\hat{p}^2\right)_{2m+1,2n+1}, \tag{2.46d}$$
$$Q^{(4e)}_{m,n} \equiv \left(\hat{q}^4\right)_{2m,2n}, \tag{2.46e}$$
$$Q^{(4o)}_{m,n} \equiv \left(\hat{q}^4\right)_{2m+1,2n+1}. \tag{2.46f}$$

The nonzero elements of the above quantities are

$$Q^{(2e)}_{n,n} = \tfrac{1}{2}(4n+1), \tag{2.47a}$$
$$Q^{(2e)}_{n,n+1} = Q^{(2e)}_{n+1,n} = \tfrac{1}{2}\sqrt{(2n+1)(2n+2)}, \tag{2.47b}$$
$$Q^{(2o)}_{n,n} = \tfrac{1}{2}(4n+3), \tag{2.47c}$$
$$Q^{(2o)}_{n,n+1} = Q^{(2o)}_{n+1,n} = \tfrac{1}{2}\sqrt{(2n+2)(2n+3)}, \tag{2.47d}$$
$$Q^{(4e)}_{n,n} = \tfrac{1}{4}(24n^2+12n+3), \tag{2.47e}$$
$$Q^{(4e)}_{n,n+1} = Q^{(4e)}_{n+1,n} = \tfrac{1}{4}(8n+6)\sqrt{(2n+1)(2n+2)}, \tag{2.47f}$$
$$Q^{(4e)}_{n,n+2} = Q^{(4e)}_{n+2,n} = \tfrac{1}{4}\sqrt{(2n+1)(2n+2)(2n+3)(2n+4)}, \tag{2.47g}$$
$$Q^{(4o)}_{n,n} = \tfrac{1}{4}(24n^2+36n+15), \tag{2.47h}$$
$$Q^{(4o)}_{n,n+1} = Q^{(4o)}_{n+1,n} = \tfrac{1}{4}(8n+10)\sqrt{(2n+2)(2n+3)}, \tag{2.47i}$$
$$Q^{(4o)}_{n,n+2} = Q^{(4o)}_{n+2,n} = \tfrac{1}{4}\sqrt{(2n+2)(2n+3)(2n+4)(2n+5)}, \tag{2.47j}$$

with further $P^{(2p)}_{nn} = Q^{(2p)}_{nn}$, $P^{(2p)}_{n,n+1} = P^{(2p)}_{n+1,n} = -Q^{((2p))}_{n,n+1}$ for parity $p = e$ or o. Note that all odd matrix elements can be obtained by shifting $n \to n + \frac{1}{2}$ in the expressions for the even ones.

2.4 Schrödinger Wave Formulation (1926)

The Schrödinger formulation of quantum mechanics [30] was much motivated by a "phase-wave" description of matter, developed in the thesis of Louis de Broglie shortly before [31]. According to the Einstein hypothesis, an electromagnetic wave of frequency ν, moving with the speed of light c, were associated "particles" (quanta) of energy $E = h\nu$. By extending this relation to massive particles, de Broglie postulated the relation

$$E = h\nu = \frac{mc^2}{\sqrt{1-(v/c)^2}}. \tag{2.48}$$

By identifying v with the wave propagation velocity, de Broglie found the wave to have a phase

$$\nu\left(t - vx/c^2\right).$$

I.e, a *de Broglie wavelength* λ_{dB} such that

$$1/\lambda_{\mathrm{dB}} = \nu v/c^2 = (mv/h)/\sqrt{1-(v/c)^2} = p/h.$$

Later one would say that a single point particle with energy E and momentum $\boldsymbol{p}$ is associated with a matter wave

$$\psi \sim \mathrm{e}^{-\mathrm{i}(Et - \boldsymbol{p}\cdot\boldsymbol{r})/\hbar}, \tag{2.49}$$

although no formula can be found in de Broglie's thesis. Further, no wave equation is postulated, but de Broglie pointed out that a quantization condition like (2.25) could be understood as a condition for constructive interference.

In reference [30] Schrödinger mainly discusses the time-independent equation for non-relativistic particles,

$$\left[-\frac{\hbar^2}{2m}\boldsymbol{\nabla}^2 + V(\boldsymbol{q})\right]\psi(\boldsymbol{q}) = E\psi(\boldsymbol{q}), \tag{2.50}$$

but mentions that a time dependent equation can be obtained by making a replacement $E\,\psi(\boldsymbol{q}) \to \pm\mathrm{i}\hbar\frac{\partial}{\partial t}\psi(\boldsymbol{q},t)$. At that time it also seemed unclear how the wave function for a system of more that one particle should

described. I.e, is the wave function a quantity in physical space (like the electromagnetic fields), so that waves from different particles must be added?

It was soon realized that it had to be function in *configuration space*, so that a system of two particles is described by a wave function $\psi(\boldsymbol{q}_1, \boldsymbol{q}_2, t)$. From later viewpoints (the Schrödinger picture) the wavefunction $\psi(\cdot, t)$ describes the state of a mechanical system in a time dependent way, with physical observables $\mathcal{O}$ being constant in time (unless they are explicitly time dependent). This is analogous to a classical description in terms of a time dependent probability distribution $\rho(\boldsymbol{q}, \boldsymbol{p}; t)$ (see section 2.1.3). This is in contrast to matrix mechanics (the Heisenberg picture), where the system is described by a time constant state vector ψ_n, as discussed in section 2.3, and time dependent observables $\hat{\mathcal{O}} = \hat{\mathcal{O}}(t)$.

In other words, the focus is shifted from the dynamics of the "measurable quantities" (observables) to the dynamics of wave function $\psi(\boldsymbol{q}, t)$ describing the state of the system, with expectation values of observables defined as

$$\langle \mathcal{O}(t) \rangle = \int d\boldsymbol{q}\, \psi(\boldsymbol{q}, t)^*\, \mathcal{O}\, \psi(\boldsymbol{q}, t). \tag{2.51}$$

Here $\boldsymbol{q}$ denotes the complete set of (generalized) coordinates. It was realized that the Schrödinger wave function formulation was (essentially) equivalent to the Heisenberg matrix formulation, only with different representations of $\boldsymbol{q}$ and $\boldsymbol{p}$,

$$\boldsymbol{q} : \psi(\boldsymbol{q}, t) \to \boldsymbol{q}\, \psi(\boldsymbol{q}, t), \tag{2.52a}$$

$$\boldsymbol{p} : \psi(\boldsymbol{q}, t) \to -\mathrm{i}\hbar \boldsymbol{\nabla}_q\, \psi(\boldsymbol{q}, t). \tag{2.52b}$$

One finds that

$$[q_j, p_k]\ \psi(\boldsymbol{q}, t) = -i\hbar \left(q_j \frac{\partial}{\partial q_k} - \frac{\partial}{\partial q_k} q_j \right) \psi(\boldsymbol{q}, t) = \mathrm{i}\hbar \delta_{jk} \psi(\boldsymbol{q}, t),$$

for all $\psi(\boldsymbol{q}, t)$. Hence, the canonical commutation relation

$$[q_j, p_k] = \mathrm{i}\hbar \delta_{jk} \tag{2.53}$$

is also obeyed in the Schrödinger formulation. The two formulations are considered equivalent, although there are some practical differences: In the Heisenberg formulation it is (in principle) straightforward to consider expectation values of observables at different times,

$$\begin{aligned} &\left\langle \mathcal{O}^{(1)}(t_1)\, \mathcal{O}^{(2)}(t_2) \cdots \mathcal{O}^{(N)}(t_N) \right\rangle \\ &= \sum_{m_0 m_1 \cdots m_N} \psi^*_{m_0} \mathcal{O}^{(1)}_{m_0 m_1}(t_1)\, \mathcal{O}^{(2)}_{m_1 m_2}(t_2) \cdots \mathcal{O}^{(N)}_{m_{N-1} m_N}(t_N)\, \psi_{m_N}. \end{aligned}$$

This seems more cumbersome to express in the Schrödinger picture. On the other hand, the Schrödinger formulation provide unambiguous results when one transforms from cartesian coordinates to other coordinate systems. As example, consider a transformation to spherical coordinates

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} r\sin\theta\cos\phi \\ r\sin\theta\sin\phi \\ r\cos\theta \end{pmatrix}, \tag{2.54}$$

which implies that

$$\begin{pmatrix} \partial_r \\ (1/r)\,\partial_\theta \\ (1/r\sin\theta)\,\partial_\phi \end{pmatrix} = \begin{pmatrix} \sin\theta\cos\phi & \sin\theta\sin\phi & \cos\theta \\ \cos\theta\cos\phi & \cos\theta\sin\phi & \sin\theta \\ -\sin\phi & \cos\phi & 0 \end{pmatrix} \begin{pmatrix} \partial_x \\ \partial_y \\ \partial_z \end{pmatrix} \tag{2.55a}$$

and

$$\begin{pmatrix} \partial_x \\ \partial_y \\ \partial_z \end{pmatrix} = \begin{pmatrix} \sin\theta\cos\phi & \cos\theta\sin\phi & -\sin\phi \\ \sin\theta\sin\phi & \cos\theta\sin\phi & \cos\phi \\ \cos\theta & \sin\theta & 0 \end{pmatrix} \begin{pmatrix} \partial_r \\ (1/r)\,\partial_\theta \\ (1/r\sin\theta)\,\partial_\phi \end{pmatrix}. \tag{2.55b}$$

It is unambigous (but tedious) to work out from this that

$$\boldsymbol{\nabla}^2 = \partial_r^2 + (2/r)\partial_r + (1/r^2)\left[\partial_\theta^2 + \cot\theta\,\partial_\theta + (1/\sin^2\theta)\,\partial_\phi^2\right]. \tag{2.55c}$$

In comparison, a classical canonical transformation gives

$$p_x^2 + p_y^2 + p_z^2 = p_r^2 + (1/r^2)\left[p_\theta^2 + (1/\sin^2\theta)p_\phi^2\right].$$

There is an ambiguity in constructing the quantum operator from this, because classically $(1/r^2)p_r\,r^2p_r = p_r^2$ and $(1/\sin\theta)p_\theta\sin\theta\,p_\theta = p_\theta^2$. But in quantum mechanics all quantities do not commute; hence these equalities are no longer true.

For example, the state of a system with two particles (spin-0) can represented mathematically by a complex function in six-dimensional configuration space $\psi(\boldsymbol{q}_1, \boldsymbol{q}_2, t)$. Which evolves in time by applying Schrödinger equation as

$$\begin{aligned} \mathrm{i}\hbar\frac{\partial\psi(\boldsymbol{q}_1,\boldsymbol{q}_2,t)}{\partial t} =& \Big[-\frac{\hbar^2}{2m_1}\boldsymbol{\nabla}_1^2\psi(\boldsymbol{q}_1,\boldsymbol{q}_2,t) - \frac{\hbar^2}{2m_2}\boldsymbol{\nabla}_2^2\psi(\boldsymbol{q}_1,\boldsymbol{q}_2,t) \\ &+ V(\boldsymbol{q}_1,\boldsymbol{q}_2)\,\psi(\boldsymbol{q}_1,\boldsymbol{q}_2,t)\Big], \end{aligned} \tag{2.56}$$

where m_1 and m_2 are the particles masses, and $V(\boldsymbol{q}_1, \boldsymbol{q}_2)$ is the classical potential energy function.

In another form, we may use the mathematical representation of momentum space, which is in six-dimension

$$\tilde{\psi}(\boldsymbol{p}_1,\boldsymbol{p}_2,t)=\int\frac{d^3q_1}{(2\pi\hbar)^{3/2}}\int\frac{d^3q_2}{(2\pi\hbar)^{3/2}}\,\mathrm{e}^{-\mathrm{i}(\boldsymbol{p}_1\cdot\boldsymbol{q}_1+\boldsymbol{p}_2\cdot\boldsymbol{q}_2)/\hbar}\,\psi(\boldsymbol{q}_1,\boldsymbol{q}_2,t),\quad(2.57)$$

where its evolution in time is

$$\begin{aligned}\mathrm{i}\hbar\frac{\partial}{\partial t}\,\tilde{\psi}(\boldsymbol{p}_1,\boldsymbol{p}_2,t)=\Bigg[&\left(\frac{\boldsymbol{p}_1^2}{2m_1}+\frac{\boldsymbol{p}_2^2}{2m_2}\right)\tilde{\psi}(\boldsymbol{p}_1,\boldsymbol{p}_2,t)\\&+\int d^3p_1'\int d^3p_2'\,\tilde{V}(\boldsymbol{p}_1',\boldsymbol{p}_2')\,\tilde{\psi}(\boldsymbol{p}_1+\boldsymbol{p}_1',\boldsymbol{p}_2+\boldsymbol{p}_2',t)\Bigg],\end{aligned}\quad(2.58)$$

such that $\tilde{V}(\boldsymbol{p}_1,\boldsymbol{p}_2)$ represent the Fourier transform of the potential energy function,

$$\tilde{V}(\boldsymbol{p}_1,\boldsymbol{p}_2)=\int\frac{d^3q_1}{(2\pi\hbar)^3}\int\frac{d^3q_2}{(2\pi\hbar)^3}\,\mathrm{e}^{\mathrm{i}(\boldsymbol{p}_1\cdot\boldsymbol{q}_1+\boldsymbol{p}_2\cdot\boldsymbol{q}_2)/\hbar}\,V(\boldsymbol{q}_1,\boldsymbol{q}_2).\quad(2.59)$$

2.4.1 Discrete Schrödinger equation

In section 2.3 we considered the possibility of replacing the operators $\hat{q}$ and $\hat{p}$ in equation (2.31) by finite $(N\times N)$ matrices,

$$\hat{q}\to\hat{q}_N,\quad \hat{p}\to\hat{p}_N.$$

Since $\hat{q}_N$ is a real symmetric matrix it can be diagonalized by a real orthogonal matrix S, $\hat{q}_N=S^{-1}\,\bar{q}_N\,S$, where $\bar{q}_N$ is real and diagonal. Hence for any function $V(q)$ one has

$$V(\hat{q}_N)=S^{-1}\,V(\bar{q}_N)\,S.\quad(2.60)$$

A corresponding rewriting of the kinetic energy becomes

$$T=\frac{1}{2m}\hat{p}_N^2=\frac{1}{2m}S^{-1}\bar{p}_N^2\,S,\quad(2.61)$$

where T is a symmetric, but non-diagonal matrix. This leads to a general discrete Schrödinger representation, with a general Hamiltonian

$$H=\frac{1}{2m}\bar{p}_N^2+V(\bar{q}_N),\quad(2.62)$$

and a diagonal position operator. It is further quite straightforward to extend this to higher dimensions D. Dropping N from the explicit notation,

$$H=\frac{1}{2m}\bar{\boldsymbol{p}}^2+V(\bar{\boldsymbol{q}}).\quad(2.63)$$

The advantage is that only the diagonal components of $V(\bar{\boldsymbol{q}})$ is nonzero, and need to be stored (alltogether N^D componenents). Further, we may write

$$\bar{\boldsymbol{p}}^2 = \sum_{j=1}^{D} \bar{p}_j^2 \tag{2.64}$$

with each $\bar{p}_j^2$ a symmetric $N \times N$ matrix. This means that the Hamiltonian at most requires the storage of $DN(N+1)/2 + N^D$ components, instead of $N^D(N^D+1)/2$.

2.5 Dirac Formulation (1927)

Some of the information in this subject taken from the Wikipedia article [32].

It's known also as interaction picture. It's a sort of intermediary between the Schrödinger picture (see section 2.4) and the Heisenberg picture (see section 2.3) as both the quantum states $|\psi\rangle$ and the operators $\hat{\mathcal{O}}$ carry time dependence.[33, 34]

$$|\psi\rangle = |\psi(t)\rangle \tag{2.65}$$

$$\hat{\mathcal{O}} = \hat{\mathcal{O}}(t) \tag{2.66}$$

This formulation is useful as it gives ability to construct the solution to the many body schrödinger equation of two parts solution. A part to the free particle problem together with unknown interaction part.

$$H_s = H_{0,s} + H_{1,s} \tag{2.67}$$

where the free parts $H_{0,s}$ have to be solvable, while $H_{1,s}$ contain the system perturbation parts. Which is not easy to solve.
Some problem have explicit time-dependent Hamiltonian (for example, if the quantum system placed within an electric field that varies in time) which considered as an advantage (including the explicity time-dependent terms with $H_{1,s}$ part, leaving the free part $H_{0,s}$ time-independent).
The state vectors of Dirac picture defined as transformation of the Schrödinger states $|\psi_s(t)\rangle$

$$|\psi_I(t)\rangle = e^{iH_{0,s}t/\hbar} |\psi_s(t)\rangle \,, \tag{2.68}$$

while the operators are transformed similarly to the Heisenberg operators

$$\hat{\mathcal{O}}_I(t) = e^{i\hat{H}_{0,s}t/\hbar}\, \hat{\mathcal{O}}_s\, e^{-i\hat{H}_{0,s}t/\hbar}. \tag{2.69}$$

Where $\hat{\mathcal{O}}_s$ is not time dependent unless the operator has "explicit time dependence".
By considering that the interaction picture counterparts to the Hamiltonian operator of Schrödinger picture.

$$\begin{aligned}\hat{H}_{0,I}(t) &= e^{i\hat{H}_{0,s}t/\hbar}\ \hat{H}_{0,s}\ e^{-i\hat{H}_{0,s}t/\hbar} \\ &= \hat{H}_{0,s}.\end{aligned} \tag{2.70}$$

For the perturbation Hamiltonian we have,

$$\hat{H}_{1,I}(t) = e^{i\hat{H}_{0,s}t/\hbar}\ \hat{H}_{1,I}\ e^{-i\hat{H}_{0,s}t/\hbar} \tag{2.71}$$

where the perturbation Hamiltonian becomes a time-dependent unless $[H_{1,s}, H_{0,s}{=}0]$.

And by transforming the Schrödinger equation into the Dirac picture we get the time-evolution of states,

$$i\hbar\frac{d}{dt}\left|\psi_I(t)\right\rangle = H_{1,I}(t)\left|\psi_I(t)\right\rangle. \tag{2.72}$$

Where it's clear that the state vector $|\psi_I(t)\rangle$ evolve in time according to the interaction term only. While the operators $\hat{\mathcal{O}}_s$ has no explicit time dependence,

$$i\hbar\frac{d}{dt}\hat{\mathcal{O}}_I(t) = \Big[\hat{\mathcal{O}}_I(t), H_0\Big]. \tag{2.73}$$

2.6 Bohm Pilot Wave Theory (1927)

Some of the information in this subject taken from the Wikipedia article [35, 36].

The pilot wave interpretation combine both the classical and wavefunction mathematical representation. While the first mechanics represent a system of one electron and one proton (spin-0) by two points tracing out trajectories in three-dimensional physical space and the other formulation represent it by a complex-valued wavefunction which evolve in six-dimensional configuration space (2.56), the pilot wave mechanics represent this system mathematically by the wavefunction (called pilot wave) via a guiding equation which provide information telling the two points how to move.

The evolution of the wave function over time is given by Schrödinger's equation,

$$i\hbar\frac{\partial\psi}{\partial t} = (-\frac{\hbar^2}{2m}\bigtriangledown^2 + V)\psi. \tag{2.74}$$

The Bohm complex wave function for this system can be represented as

$$\psi(\boldsymbol{q}_1, \boldsymbol{q}_2, t) = R(\boldsymbol{q}_1, \boldsymbol{q}_2, t)\, e^{iS(\boldsymbol{q}_1, \boldsymbol{q}_2, t)/\hbar}, \tag{2.75}$$

where $R(\boldsymbol{q}_1, \boldsymbol{q}_2, t)$ is the real magnitude term and $e^{iS(\boldsymbol{q}_1, \boldsymbol{q}_2, t)/\hbar}$ is the phase functions term.

By plugging equation(2.75) into schrödinger equation(2.74) we get first the continuity equation for the probability density P[37],

$$\frac{\partial P}{\partial t} + \frac{1}{m_1}\boldsymbol{\nabla}_1.(P\boldsymbol{\nabla}_1 S) + \frac{1}{m_2}\boldsymbol{\nabla}_2.(P\boldsymbol{\nabla}_2 S) = 0, \tag{2.76}$$

where[6]

$$P(\boldsymbol{q}_1, \boldsymbol{q}_2, t) = R^2(\boldsymbol{q}_1, \boldsymbol{q}_2, t) \tag{2.77}$$

and

$$\frac{1}{m_i}\boldsymbol{\nabla}_i S \quad \text{for } i = 1, 2 \tag{2.78}$$

represent the velocity field of the system particles which is defined by the guidance equation[7].

And we get a modified Hamilton-Jacobi equation for the action S,

$$\frac{\partial P}{\partial t} = -\frac{(\boldsymbol{\nabla}_1 S)^2}{2m_1} - \frac{\boldsymbol{\nabla}_2 S)^2}{2m_2} - V(\boldsymbol{q}_1, \boldsymbol{q}_2) - Q(\boldsymbol{q}_1, \boldsymbol{q}_2, t) \tag{2.79}$$

where $Q(\boldsymbol{q}_1, \boldsymbol{q}_2, t)$ represent the state-dependent "quantum potential"

$$Q(\boldsymbol{q}_1, \boldsymbol{q}_2, t) = -\frac{\hbar^2}{2m_1}\frac{\boldsymbol{\nabla}_1^2 R}{R} - \frac{\hbar^2}{2m_2}\frac{\boldsymbol{\nabla}_2^2 R}{R}. \tag{2.80}$$

Based on this the two point particles will move with accelerations

$$m_1\frac{d\boldsymbol{v}_1}{dt} = -\boldsymbol{\nabla}_1 V - \boldsymbol{\nabla}_1 Q \quad \text{and} \quad m_2\frac{d\boldsymbol{v}_2}{dt} = -\boldsymbol{\nabla}_2 V - \boldsymbol{\nabla}_2 Q \tag{2.81}$$

From the last term in equation(2.81) we can understand that the force is given by both the gradient of the classical and quantum potential as well.

We can arrive to the classical limit of the Bohm pilot wave formulation by neglecting the quantum potential Q. This will reduce equation(2.79) to the Hamilton-Jacobi of a classical point particle[38].

[6] Since the probability density defined as $P(\boldsymbol{q}, t) = |\psi(\boldsymbol{q}, t)|^2$

[7] The guiding equation for many particles k

$$\frac{d\boldsymbol{q}_k}{dt}(t) = \frac{\hbar}{m_k} Im\Big(\frac{\boldsymbol{\nabla}_k \psi}{\psi}\Big)(\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_N, t),$$

where the velocity firld depends on the actual positions of all the N particles in the universe.

2.7 Wigner Phase Space Formulation (1932)

An alternative formulation which is not very popular. It's probably equivalent to the schrödinger formulation of quantum mechanics[8], which is the standard formulation.

This formulation was given in 1932 in a paper on the Quantum Correction for Thermodynamic Equilibrium.[39] Wigner was not looking for finding another formulation of quantum mechanics, but rather in finding a quantum correction (*Boltzmann equation which is the main equation in the classical statistical mechanics*).

For a single particle in one dimension, the Wigner phase-space distribution function, which is given in terms of Quasi-distribution functions, is

$$W(x,p,t) = \frac{1}{2\pi\hbar}\int_{-\infty}^{+\infty} \psi^*(x-\frac{1}{2}y,t)\psi(x+\frac{1}{2}y,t)e^{-ipy/\hbar}\,dy. \tag{2.82}$$

A quasi-distribution is not a probability distribution since it has a negative peak at some places. But if we integrates over impulses we get real probability distribution of positions.

$$\int_{-\infty}^{+\infty} W(x,p,t)dy = |\psi(x,t)|^2. \tag{2.83}$$

And if we integrats over positions we get real probability distribution of impulses.

$$\int_{-\infty}^{+\infty} W(x,p,t)dx = |\widetilde{\psi}(p,t)|^2. \tag{2.84}$$

The time development of equation (2.82) is

$$\frac{\partial W(x,p,t)}{\partial t} = -\frac{p}{m}\frac{\partial W(x,p,t)}{\partial x}\int_{-\infty}^{+\infty} K(x,p')\; W(x,p+p',t)\,dp' \tag{2.85}$$

where the kernel $K(x,p)$ is

$$K(x,p) = \frac{1}{2\pi\bar{h}}\int_{-\infty}^{+\infty}\left[V(x-\frac{1}{2}y) - V(x+\frac{1}{2}y)\right]\sin(\frac{py}{\hbar})\,dy. \tag{2.86}$$

It's difficult to solve equation (2.85) even numerically.

Where a quasi distribution function defined over the phase space and it's 6^{th} dimensional space. And if we have n particles system, its phase space becomes a $(6\times n)$ dimensional space. Which is mathematically very difficult problem.

[8]Mathematically, Wigner formulation and Schrödinger formulation are equivalent. By using Wigner-Weyl transformation, we can start from one formulation to get the other corresponding one.

2.8 The Path Integral Formulation (1948)

Some of the information in this subject taken from the Wikipedia article [40].

It is an approach to quantum mechanics by Feynman that was already described by Dirac early (see section 2.5). This formulation shifts the focus from "state" to "Transition probability".

The basic idea behind the path integral is that to calculate the quantum transition amplitude for a particle moved from a particular position $\boldsymbol{q}_i$ at a particular time t_i to a different position $\boldsymbol{q}_f$ at a later time t_f. Where we have to consider all the possible classical space-time paths that the particle can take from the initial to the final state (calculate the classical action $S = \int (Lagrangian) dt$ for each path) assign each path a "transition amplitude " proportional to $e^{iS/\hbar}$.

And to calculate the total amplitude for the particle movement we have to add up all the complex amplitudes over all the paths the particle can take through it's journey or by other words we take the path integral as this movement will have a continuum of paths.

$$\psi(t_1, \boldsymbol{q}_1) = \int e^{\frac{iS(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\hbar}} dq(t)\ \psi(t_0, \boldsymbol{q}_0) \tag{2.87}$$

where the asction is

$$S = \int dt \big[\frac{1}{2} m\dot{q}^2 - V(q)\big]. \tag{2.88}$$

Integrate (2.87) over all possible paths from $(t_0, \boldsymbol{q}_0)$ to $(t_1, \boldsymbol{q}_1)$. Here

$$t \to -i\tau, \quad dt = id\tau$$

and,

$$\Big(\frac{dq}{dt}\Big)^2 = \Big(\frac{d\sigma}{dt}\Big)^2 \Big(\frac{dq}{d\sigma}\Big)^2 = -\Big(\frac{dq}{d\tau}\Big)^2$$

.

Then the action S in equation (2.88) become,

$$S_E = \int d\tau \Big[\frac{1}{2}\Big(\frac{dq}{d\tau}\Big)^2 + V(q)\Big]. \tag{2.89}$$

And equation (2.87),

$$\psi(t_1, \boldsymbol{q}_1) = \int e^{\frac{-iS_E}{\hbar}} dq(t)\ \psi(t_0, \boldsymbol{q}_0). \tag{2.90}$$

The standard way to interpret this amplitude is by taken it's square absolute value, which we call it the probability, such that

$$\int \left|\psi(t,q)\right|^2 dq = 1 \tag{2.91}$$

for $V = 0$ (Brownian motion).Comparing this formulation to Bohm formulation section 2.6 we find that path integral formulation is intrinsically relativistic and it does not require any inertial frame, while Bohm mechanics need the fram to applied to a relativistic system[41, 42].

One of the advantages of this formulation is that we can arrive to the classical limit by $\hbar \to 0$, as the weight factor $e^{iS/\hbar}$ will oscillate very rapidly. By other words, in small $\hbar$ limit the path integral dominates by the classical trajectory.

Chapter 3

Spectral theory

Some of the information in this subject taken from the Wikipedia article [43].

In physical models which can be described by linear equations, such as quantum mechanics, it is common and convenient to describe the time evolution as a superposition of fundamental modes of oscillations, each mode undergoing periodic motion with a given frequency. Examples can be the determination of the natural oscillation frequencies of an harmonic string or a drum, or the energy eigenvalues of quantum systems. From this pont of view spectral theory is the mathematical tool to determine the fundamental frequencies (eigenvalues), and the corresponding modes (eigenvectors).

But on the mathematical and technical level, spectral theory is not dependent on such a physical perspective. In particular from a practical point of view, where only systems with a finite numbers of modes can be handled numerically, it can be considered to be a subset of linear algebra. As such it consists of a collection of theoretical concepts and theorems, together with a set of theoretical and practical methods to find a discrete set of eigenvalues and eigenvectors of finite matrices.

3.1 Spectral representation

This reference [44] was useful of writing this part.

A $N \times N$ matrix A can be *completely diagonalized* if we can find N eigenvalues D_n, each with a corresponding left eigenvector $\chi^{(n)}$, and corresponding right eigenvector $\varphi^{(n)}$, such that

$$\chi^{(n)} A = D_n \chi^{(n)}, \qquad A \varphi^{(n)} = \varphi^{(n)} D_n, \quad \text{for } n = 1, 2, \ldots, N. \tag{3.1}$$

By using both of these relations, and subtracting, we find that

$$0 = \chi^{(m)} A\, \varphi^{(n)} - \chi^{(m)} A\, \varphi^{(n)} = (D_m - D_n)\, \chi^{(m)} \cdot \varphi^{(n)}.$$

I.e., when $D_m \neq D_n$ we must have that

$$\chi^{(m)} \cdot \varphi^{(n)} \equiv \sum_{k=1}^{N} \chi_k^{(m)} \varphi_k^{(n)} = 0. \tag{3.2}$$

When $D_m = D_n$ for $m \neq n$, the matrix is said to have *degenerate* eigenvalues. Then the *ortogonality condition* (3.2) can be achieved through a Gram-Schmidt ortogonalization process.[refer to an appendix] Further, the relative normalizations of left and right eigenvectors can be chosen such that $\chi^{(n)} \cdot \varphi^{(n)} = 1$. With a complete set of such *orthonormalized* left and right eigenvectors, we can define a $N \times N$ matrix S^{-1} such that its n^{th} row equals $\chi^{(n)}$, and another matrix S such that the n^{th} column in S equals $\varphi^{(n)}$. Explicitly

$$S^{-1} = \begin{pmatrix} \chi_1^{(1)} & \chi_2^{(1)} & \cdots & \chi_n^{(1)} \\ \chi_1^{(2)} & \chi_2^{(2)} & \cdots & \chi_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ \chi_1^{(n)} & \chi_2^{(n)} & \cdots & \chi_n^{(n)} \end{pmatrix}, \qquad S = \begin{pmatrix} \varphi_1^{(1)} & \varphi_1^{(2)} & \cdots & \varphi_1^{(n)} \\ \varphi_2^{(1)} & \varphi_2^{(2)} & \cdots & \varphi_2^{(n)} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_n^{(1)} & \varphi_n^{(2)} & \cdots & \varphi_n^{(n)} \end{pmatrix}.$$

The orthonormalization condition can then be written in a matrix form

$$\left(S^{-1}S\right)_{mn} = \sum_{k=1}^{N} \chi_k^{(m)} \varphi_k^{(n)} = \delta_{mn}. \tag{3.3a}$$

This means that $S^{-1}S = I$, where I is the $N \times N$ unit matrix, so that S^{-1} is the (left) inverse of S (as the notation suggests). But since the left and right inverse of finite-dimensional matrices are equal, we must also have

$$\left(SS^{-1}\right)_{jk} = \sum_{n=1}^{N} \varphi_j^{(n)}\, \chi_k^{(n)} = \delta_{jk}. \tag{3.3b}$$

This is the *completeness relation* for the eigenvectors. By introducing Dirac notation we can rewrite the *completeness relation* as,

$$\sum_n |n\rangle \langle n| = 1,$$

and the *orthogonality* as,

$$\langle n|m\rangle = \delta_{nm}.$$

And as we know the relation between χ and φ is $\chi_j = \sum_n C_n \varphi_j^n$ or, we can rewrite it as,

$$|\chi\rangle = \sum_n |n\rangle \langle n|\chi\rangle,$$

where $|\chi\rangle \equiv \chi_k$, $|n\rangle \equiv \varphi_j^n$ and $\langle n|\chi\rangle \equiv C_n$.

The previous definitions can be used to decompose the matrix A into the form

$$A = S^{-1} D S$$

from which we may evaluate functions of A as

$$f(A) = S^{-1} f(D) S$$

Here D is a diagonal matrix, $D = \mathrm{diag}(D_1, D_2, \ldots, D_N)$.

The generalization to systems with infinitely many modes leads to a much broader theory of linear operators [45], with extended possibilities for spectra and (generalized) eigenmodes. One may say that the spectral theorem identifies a class of linear operators that can be represented by multiplication operators, after a suitable change of basis. This is as simple as a linear operator can be.

At this point it should be pointed out that not all matrices can be completely diagonalized. The problem is related to the occurence of degenerate eigenvalues. To each eigenvalue, not counting degeneracies, there is at least one left and right eigenvector. But not necessarily equally many as the numer of degenerate eigenvalues. The latter is called the *algebraic multiplicity*, while the number of linearly independent eigenvectors corresponding to the eigenvalue is called the *geometric multiplicity*. I.e, the geometric multiplicity is equal or smaller than the algebraic multiplicity, but never smaller than 1.

3.2 Diagonalization of general matrices

Consider a general $n \times n$ matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix}, \tag{3.4}$$

which we want to diagonalize. This means that we must find a matrix P such that $P^{-1}AP = D$, where D diagonal. I.e.,

$$P^{-1}AP = D = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{pmatrix}. \tag{3.5}$$

Equation (3.5) can also be written as $AP = PD$, or written explicitly:

$$\sum_k A_{ik}P_{kj} = P_{ik}D_{kj} = P_{ij}\lambda_j. \tag{3.6}$$

In the last equality we have used that D is diagonal, $D_{kj} = \delta_{kj}\lambda_j$. This means that the vector $v^{(j)}$ with components $v_i^{(j)} \equiv P_{ij}$ is a right eigenvector of A with eigenvalue λ_j:

$$\sum_k A_{ik}v_k^{(j)} = \lambda_j\, v_i^{(j)}, \quad \text{or } A\, v^{(j)} = \lambda_j v^{(j)}, \text{ for } j = 1, 2, \dots n, \tag{3.7}$$

in matrix notation. This means that the diagonalizing matrix P consists of the right eigenvectors of A,

$$P = \left(v^{(1)}, v^{(2)}, \dots, v^{(n)}\right). \tag{3.8}$$

We can rewrite equation (3.7) as $(A - \lambda_j I)v^{(j)} = 0$, where I is the unit matrix. This equation has a nonzero solution only when there is a λ such that

$$\det(A - \lambda I) = 0. \tag{3.9}$$

As can be seen by expanding the determinant, this is a n^{th} order algebraic equation for λ. Hence, counting degeneracies, the eigenvalue equation (3.9) has n roots.

3.2.1 The case of 2×2 matrices

To illustrate the general procedure, and the exceptional cases which may arise for arbitrary square matrices, consider the case of a 2×2 matrix,

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22}. \end{pmatrix} \tag{3.10}$$

To find an eigenvalue and a right eigenvector of A means that we must find a number λ and a non-zero vector $\boldsymbol{v}$, such that

$$A\boldsymbol{v} = \lambda\boldsymbol{v}, \quad \text{or} \quad (A - \lambda I)\boldsymbol{v} = \mathbf{0}, \tag{3.11}$$

where I is the 2×2 unit matrix. If $A - \lambda I$ is invertible, we can just solve equation (3.11), with the result that $\boldsymbol{v} = (A - \lambda I)^{-1}\,\mathbf{0} = \mathbf{0}$, in contradiction to the requirement that $\boldsymbol{v}$ be non-zero. Hence, to have an eigenvector the matrix $(A - \lambda I)$ cannot be invertible, which is the case if and only if its determinant is 0,

$$\det(A - \lambda I) = (a_{11} - \lambda)(a_{22} - \lambda) - a_{12}\,a_{21} = 0. \tag{3.12}$$

This is the *eigenvalue equation*, which in this case is a second order equation for λ. Hence, counting degeneracies, we find two solutions,

$$\lambda_{\pm} = \frac{1}{2}\left[(a_{11} + a_{22}) \pm \sqrt{(a_{11} - a_{22})^2 + 4a_{12}a_{21}}\right]. \tag{3.13}$$

Consider first the case of two degenerate solutions, $\lambda_+ = \lambda_-$. This occur when the square root in (3.13) vanishes; i.e., when $(\mathrm{Tr}A)^2 = 4\,\mathrm{det}A$. In this case we find the eigenvector equation to be

$$(A - \lambda I)\,\boldsymbol{v} = \begin{pmatrix} \sqrt{-a_{12}\,a_{21}} & a_{12} \\ a_{21} & -\sqrt{-a_{12}\,a_{21}} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = 0. \tag{3.14}$$

This has two linearely independent solutions only when $a_{12} = a_{21} = 0$. I.e., when A is proportional to the unit matrix (in which case it is already diagonal). Otherwise the *geometric multiplicity* (the number of linearly independent eigenvectors corresponding to an eigenvalue) is smaller than the *algebraic multiplicity* (the number of degenerate eigenvalues).

3.3 Diagonalization of Hermitian matrices

This reference [46] was useful of writing this part.

We are always able to diagonalize hermitian matrices, and more generally normal matrices (i.e. Commutator with their hermitian conjugate). Real hermitian matrices are symmetrical, that means, they are identical to the hermitian conjugate. Diagonalisation matrices are unitary (orthogonal of real matrices).

Consider a hermitian $n \times n$ matrix A

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix}, \tag{3.15}$$

where $a_{ji} = a_{ij}^*$.

We will first prove that it's eigenvalues are real and the eigenvectors corresponding to different eigenvalues are orthogonal.
Denote $\psi^(j)$ as the eigenvectors to matrix A, with λ_j as the matrix eigenvalues, i.e.

$$\sum a_{ik}\psi_k^{(j)} = \lambda_j \psi_i^{(j)} \tag{3.16}$$

where both $\psi^(j)$ and λ_j a priori may be complex, and

$$\psi^{(j)} = \left(\psi_1^j, \psi_2^j, \psi_3^j, \ldots\right)^T. \tag{3.17}$$

By multiplying equation (3.16) by $\psi_i^{*(j)}$, and summation over i, we get

$$\sum_{ik} a_{ik}\psi_i^{*(j)}\psi_k^{(j)} = \lambda_j \sum_i \psi_i^{*(j)}\psi_i^{(j)}, \tag{3.18a}$$

or

$$\lambda_j = \frac{\sum_{ik} a_{ik}\psi_i^{*(j)}\psi_k^{(j)}}{\sum \psi_i^{*(j)}\psi_i^{(j)}}. \tag{3.18b}$$

Here the number on the right hand side is real, because $a_{ik}^* = a_{ki}$,

$$\left(\sum_{ik} a_{ik}\psi_i^{*(j)}\psi_k^{(j)}\right)^* = \sum_{ik} a_{ik}^*\psi_i^{(j)}\psi_k^{(j)*} \tag{3.19a}$$

$$= \sum_{ik} a_{ki}\psi_k^{(j)*}\psi_i^{(j)}. \tag{3.19b}$$

Consider next another eigenvector ψ^q corresponding to an eigenvalue $\lambda_q \neq \lambda_i$,

$$\sum_k a_{ik}\psi_k^{(q)} = \lambda_q \psi_i^{(q)}. \tag{3.20}$$

Multiply this by $\psi_i^{(j)*}$ and, sum over i, and complex conjugate the expression

$$\left(\sum_{ik} \psi_i^{(j)*} a_{ik}\psi_k^{(q)}\right)^* = \sum_{ik} \psi_k^{(q)*} a_{ik}^*\psi_i^{(j)} \tag{3.21a}$$

$$= \sum_{ik} \psi_k^{(q)*} a_{ki}\psi_i^{(j)} = \lambda_j \sum_k \psi_k^{(q)*}\psi_k^{(j)} \tag{3.21b}$$

$$= \lambda_q \sum_i \psi_i^{(q)*}\psi_i^{(j)}. \tag{3.21c}$$

I.e.,

$$\left(\lambda_j - \lambda_q\right) \sum_k \psi_i^{*(q)} \psi_i^{(j)} = 0. \tag{3.22}$$

Since $\lambda_j - \lambda_q \neq 0$ by assumption, we must have that

$$\sum_i \psi_i^{*(a)} \psi_i^{(j)} = 0. \tag{3.23}$$

It is possible that $\lambda_j = \lambda_q$ for $j \neq q$ (degeneracy), but it can be shown that the result still holds. if the eigenvectors ψ_k^j are chosen such that

$$\sum_i \psi_k^{*j} \psi_k^j = 1, \tag{3.24}$$

then they are orthonormalized. We can rewrite equation (3.23) as

$$\sum_i \psi_k^{*j} \psi_k^q = \delta_{iq}. \tag{3.25}$$

Now, we have to show that the matrix U is unitary matrix, where the columns elements are the orthonormalized eigenfunctions ψ_i^j of the matrix A. The elements of the unitary matrix U are,

$$U_{ij} = \psi_i^j \Longleftrightarrow U_{ij}^{-1} = U_{ji}^* = \psi_j^{*(i)} \tag{3.26}$$

Hence,

$$\left(U^{-1}U\right)_{i1} = U_{ij}^{-1} U_{j1} = \psi_j^{*(i)} \psi_j^{(1)} = \delta_{i1} \tag{3.27}$$

or,

$$U^* U = 1 \Longleftrightarrow U^* = U^{-1}. \tag{3.28}$$

Which means that the matrix U is unitary. Next, we have to show that $U^{-1}AU$ is a diagonal matrix, where the diagonal elements are the eigenvalues λ_j of the matrix A.

$$\begin{aligned} &\left(U^{-1}AU\right)_{ij} = \left(U^* AU\right)_{ij} \\ &= U_{i1}^* a_{1i} U_{ij} = \psi_1^{*(i)} a_{1i} \psi_i^{(j)} = \psi_1^{*(i)} \lambda_j \psi_1^{(j)} = \lambda_j \left(\psi_1^{*(i)} \psi_1^{(j)}\right) \end{aligned} \tag{3.29}$$

In other words,

$$A^r = U^{-1} A U \tag{3.30}$$

is a diagonal matrix, where the diagonal elements are the eigenvalues of the matrix A. Hence that the two matrices A^r and A are similar for any non-singular matrix U. This means that by this unitary-similarity transformation equation (3.30) it is always possible to diagonalize a Hermitian matrix.

Chapter 4

Spectral analysis in Python

This chapter gives an introduction — by way of examples — to the Python computer language, and the numerical array and linear algebra routines in the `NumPy` and `SciPy` packages. In particular the diagonalization routines in `scipy.linalg` and `scipy.sparse.linalg`. The examples provide an overview of most coding used in the models discussed later. They are given in a form of short snippets which should be simpler to grasp than the complete program (included in appendix A).

4.1 Numerical matrix mechanics

4.1.1 Constructing matrices. Numerical matrix algebra

Consider the matrices $\hat{q}$ and $\hat{p}$ defined in equations (2.31), but restriced to the $N \times N$ matrices (which we denote as $\hat{q}_N$ and $\hat{p}_N$). By choosing units such that $m\omega = 1$ and $\hbar = 1$, first superdiagonal of $\hat{q}_N$ can be defined as in `numpy` by the the code

```
N = 10
Q11 = numpy.sqrt(numpy.arange(1, N)/2)
```

Here `arange` constructs a sequence `[1., 2., ... ,9.]` (a `numpy.array`), after which each element is divided by `2.`, and finally the square root of each element is taken. Note that `Q11` contain $N-1$ elements, not N. In general, the syntax is such that last element of a range is not included. `Q11[0]` refers to the first element of `Q11`, `Q11[-1]` refers to the last element, and `Q11[:-1]` refers to all elements except the last.

We may use `Q11` to define the action of $\hat{q}_N$ and $\hat{p}_N$ on a N-component vector `psi`, in terms of the routines

```
def q(psi):
    Psi = numpy.zeros_like(psi)
    Psi[:-1] += Q11 * psi[1:]
    Psi[1:]  += Q11 * psi[:-1]
    return Psi
```

Here the first line constructs a 0-filled array `Psi` with the same structure as `psi`. The second line multiplies the last $N-1$ elements of `psi` with the corresponding elements of `Q11` and adds the result to the first $N-1$ elements of `Psi`. Since the matrices only differ in the sign of the subdiagonal, the corresponding code for $i\hat{p}_N$ becomes

```
def ip(psi):
    Psi = numpy.zeros_like(psi)
    Psi[:-1] += Q11 * psi[1:]
    Psi[1:]  -= Q11 * psi[:-1]
    return Psi
```

Note that the code above contain no explict `for`-loops (or similar); the syntax is such that iteration over array elements is handled automatically by efficient compiled code. This is important, because native python loops would often be impractically slow.

So far no explicit matrices have been constructed; what we have are examples of the `LinearOperator` concept, which in many cases are all that is needed by the iterative routines in `scipy.sparse.linalg`. Routines in `scipy.linalg` require normal matrices, however. It is therefore useful to write a general routine which constructs a matrix from a `LinearOperator`

```
def matrix(operator):
    M = numpy.zeros((N, )*2)       # Space for matrix
    id = numpy.eye(N)              # The unit matrix
    for n in range(N):
        M[:,n] = operator(id[:,n]) # Add n'th column
    return M
```

This is an example where an explicit python `for`-loop is acceptable, several array operations are performed in each iteration. One may now construct matrix representations of `q` and `ip`, and check the commutator

```
Mq = matrix(q); Mip = matrix(ip)
C = numpy.dot(Mq, Mip) - numpy.dot(Mip, Mq)
print (C)
```

The result is (for $N = 5$):

```
[[-1.  0.  0.  0.  0.]
 [ 0. -1.  0.  0.  0.]
 [ 0.  0. -1.  0.  0.]
 [ 0.  0.  0. -1.  0.]
 [ 0.  0.  0.  0.  4.]]
```

Apart from the last element, this agrees with the canonical commutation relation, since one should have (in units where $\hbar = 1$) $[\hat{q}, \mathrm{i}\hat{p}] = \mathrm{i}^2 = -1$. But such a relation is impossible to obtain for finite matrices, since one generally must have

$$\mathrm{Tr}\,[A, B] \equiv \sum_m (AB - BA)_{mm} = \sum_{mn} (A_{mn}B_{nm} - B_{nm}A_{nm}) = 0.$$

I.e., the matrices $\hat{q}_N$ and $\hat{p}_N$ do their best to fulfill the commutation relation, by only changing the last diagonal element of the commutator, in such a way that it becomes traceless.

4.1.2 Matrix diagonalization by scipy.linalg.eigh

We may next construct a quadratic Hamiltonian $H = \left(\hat{p}_N^2 + \hat{q}_N^2\right)$,

```
H = -numpy.dot(Mip,Mip) + numpy.dot(Mq, Mq)
print (H)
```

The result (for $N = 5$) show that this matrix is already diagonal,

```
[[ 1.  0.  0.  0.  0.]
 [ 0.  3.  0.  0.  0.]
 [ 0.  0.  5.  0.  0.]
 [ 0.  0.  0.  7.  0.]
 [ 0.  0.  0.  0.  4.]]
```

but that the last eigenvalue differs from the exact result, $E_n = (2n+1)$. For a less trivial situation, we consider the anharmonic oscillator, $H = \hat{p}_N^2 + \hat{q}_N^4$. The corresponding matrix is sparse, with the only non-zero elements being $H_{m,m}$, $H_{m,m+2} = H_{m+2,m}$, and $H_{m,m+4} = H_{m+4,m}$. It can be diagonalized by the `eigh` routine from `scipy.linalg`, which computes eigenvalues and eigenvectors of real symmetric or complex hermittian matrices:

```
E, S = eigh(H)
```

Here `E` is a N-component array of eigenvalues, and `S` a $N \times N$ is an array of eigenvectors, such that `S[:, n]` is the n^{th} eigenvector, with eigenvalue `E[n]`. We can check how well this works:

```
for n in range(N):
    psi = S[:, n]; dpsi = numpy.dot(H, psi) - E[n]*psi
    print ("%.5e" % numpy.linalg.norm(dpsi, ord=numpy.inf))
```

This prints out the largest absolute component of each `dpsi` (which analytically should be zero):

```
1.08247e-15
2.44249e-15
4.70742e-16
5.32907e-15
3.55271e-15
```

This is quite close to the obtainable accuracy of double precision numbers, but becomes gradually worse as N increases. One may also check the eigenvectors for orthonormality, $S^T S = \mathbb{1}$, and completeness, $S\,S^T = \mathbb{1}$.

```
unity = numpy.dot(S.T, S)
ortherr = numpy.linalg.norm(unity - numpy.eye(N),
                            ord=numpy.inf)
unity = numpy.dot(S, S.T)
comperr = numpy.linalg.norm(unity - numpy.eye(N),
                            ord=numpy.inf)
print ("N=%4d: ortherror=%.5e  comperr=%.5e" %
(N, ortherr, comperr))
```

The results show a reasonable degradation of accuracy as N increases.

```
N=   5: ortherror=6.85948e-16  comperr=7.85608e-16
N=  20: ortherror=3.90616e-15  comperr=3.65959e-15
N=  50: ortherror=9.28002e-15  comperr=1.02578e-14
N= 200: ortherror=2.86420e-14  comperr=4.20343e-14
N= 500: ortherror=8.04324e-14  comperr=9.10659e-14
N=2000: ortherror=3.09277e-13  comperr=5.90044e-13
N=5000: ortherror=8.61635e-13  comperr=1.59215e-12
```

4.1.3 Full spectral decomposition

One may also check how the workings of a full spectral decomposition,

$$D = S^T H S$$

```
D = numpy.dot(S.T, numpy.dot(H, S))
err = numpy.linalg.norm(D - numpy.diag(E), ord=numpy.inf)
```

```
dt = time.time() - t0
print ("N=%4d: maxerror=%.5e  time=%8.5f secs" % (N, err, dt))
```

The printout shows that one may solve quite large system in reasonable time, but that the loss of numerical accuracy may be a more significant problem.

```
N=   5: maxerror=1.54613e-14  time= 0.01167 secs
N=  20: maxerror=4.07478e-12  time= 0.00418 secs
N=  50: maxerror=3.06072e-11  time= 0.00375 secs
N= 200: maxerror=1.41194e-09  time= 0.01578 secs
N= 500: maxerror=1.27761e-08  time= 0.09018 secs
N=2000: maxerror=5.29558e-07  time= 2.57153 secs
N=5000: maxerror=6.72480e-06  time=29.76773 secs
```

4.1.4 Matrix diagonalization by scipy.sparse.linalg.eigsh

For very large systems one may use routines in `scipy.sparse.linalg` to find a subset of eigenvalues, and optionally eigenvectors, using iterative methods. Without extra programming efforts this works fastest for eigenvalues of the largest magnitude, while in quantum mechanics one is usually interested in the eigenvalues of the smallest magnitude (where `eigsh` also works, but perhaps in a less impressive way).

A straightforward (but less than optimal) solution is the code snippet below, which computes the `k` eigenvalues of smallest magnitude, and their corresponding eigenvectors:

```
def p2(psi):
    chi = ip(psi)
    return -ip(chi)

def q2(psi):
    chi = q(psi)
    return q(chi)

def H(psi):
    chi = q2(psi)
    return p2(psi) + q2(chi)

A = LinearOperator((N, N), matvec=H, dtype=float)
E, S = eigsh(A, k=k, which='SM')
```

For each N we have computed the maximal violation of the eigenvalue condition,

$$\texttt{maxerror} = \max_n \|(H - E_n)\psi^{(n)}\|_\infty,$$

and the time used to find the solution. One should be aware that asking for fewer eigenvalues does not always lead to a faster solution; sometimes this may even lead to a convergence failure of `eigsh`.

```
N=  50 (k= 20): maxerror=2.62013e-12  time=  0.08374 secs
N= 200 (k= 20): maxerror=4.21337e-11  time=  0.25605 secs
N= 500 (k= 80): maxerror=4.86842e-10  time=  0.90722 secs
N=2000 (k= 80): maxerror=4.61745e-09  time= 11.17339 secs
N=5000 (k=100): maxerror=3.08765e-08  time=121.62463 secs
```

4.2 Numerical wave mechanics

In the Schrödinger picture the state of a system is described by a time dependent "wave function". In units where $\hbar = 1$,

$$\mathrm{i}\frac{d}{dt}\,|\psi(t)\rangle = H\;|\psi(t)\rangle\,, \quad |\psi(0)\rangle = |\psi_0\rangle. \tag{4.1}$$

The solution can be written

$$|\psi(t)\rangle = \mathrm{e}^{-\mathrm{i}Ht}\;|\psi_0\rangle = \sum_{mn} |m\rangle\langle m|\mathrm{e}^{-\mathrm{i}Ht}|n\rangle\langle n|\psi_0\rangle = \sum_n |n\rangle \mathrm{e}^{-\mathrm{i}E_n t}\,\langle n|\psi_0\rangle,$$

where we have inserted the completeness relation $\sum_n |n\rangle\langle n| = \mathbb{1}$ twice, with $|n\rangle$ the eigenvalue E_n eigenstate of H. The amplitude $M_0(t)$ for being in the state $|\psi_0\rangle$ at time t is

$$M_0(t) = \langle\psi_0|\psi(t)\rangle = \sum_n (\cos E_n t + \mathrm{i}\sin E_n t)\,|\langle n|\psi_0\rangle|^2 \tag{4.2}$$

This gives the survival probability

$$p_0(t) = |M_0(t)|^2 = M_{0R}^2 + M_{0I}^2, \tag{4.3}$$

where M_{0R} and M_{0I} are the real and imaginary parts of $M_0(t)$. The corresponding amplitude $M_k(t)$ for being in a state $|\psi_k\rangle$ at time t is obtained by the replacement $|\langle n|\psi_0\rangle|^2 \to \langle n|\psi_k\rangle^*\langle n|\psi_0\rangle$ in equation (4.2).

A numerical implementation can be as follows (after the matrix `H` has been generated)

```
E, S = eigh(H)
Nt = 200
t = numpy.linspace(0., 150., Nt)
Et = t.reshape(Nt,1) * E.reshape(1,N) # Broadcast mechanism
S02 = S[0,:]**2
M0R = numpy.dot(numpy.cos(Et), S02)
M0I = numpy.dot(numpy.sin(Et), S02)
P0t = M0R**2 + M0I**2
```

Note how `Et` is constructed as a two-dimensional array of size `Nt`×`N`, by multiplication of two one-dimensional arrays of sizes `Nt` and `N`, with use of the *Broadcast mechanism* in `NumPy`. Actually, this is similar to what is done analytically every time one multiplies two functions of different variables, like $h = f(t)\, g(E)$. Here one must provide some structure information, by redefining a N-dimensional vector as a $(1 \times N)$-dimensional matrix — which is different from a $(N \times 1)$-dimensional matrix. The numerical data is not changed or moved by this, only the information of how the data should be interpreted.

Chapter 5

The harmonic oscillator

In this chapter we make a detailed discussion of classical harmonic oscillators. The motivation for this was to simulate the Langevin equation for the dynamics of "particles" in a fluctuating environment. In classical physics such fluctuations are usually considered to be of thermal origin, but this is similar to fluctuations of quantum origin. In the stochastic quantization method of Parisi and Wu [47] a similar coupling between dynamical variables and randomly fluctuating "force fields" was introduced.

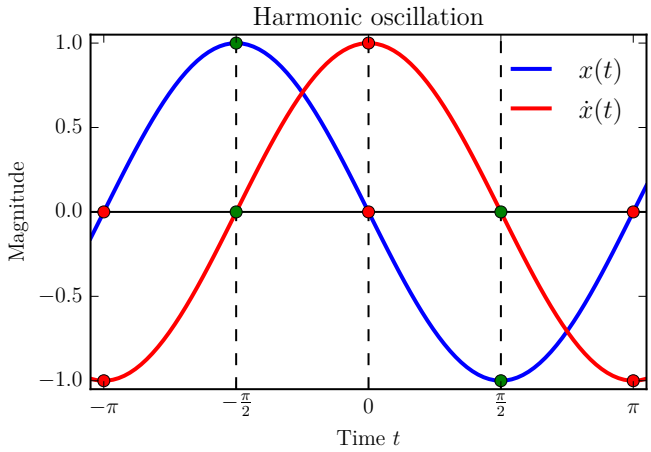


Figure 5.1: *This figure shows the motion (position and velocity) of a planar pendulum (i.e., one-dimensional motion) in the small angle approximation.*

5.1 Examples of harmonic oscillating systems

One of the most encountered systems in physics is the harmonic oscillator. This has exactly known analytical solutions, both in classical and quantum physics. It occurs very often in physics and elsewhere, because many systems may be considered close to some equilibrium configuration. Then, to first order in deviation from equilibrium the dynamics is usually determined by linear equations of motion. Among systems in our world which often can be described as one or more harmonic oscillators one finds the planar pendulum, atomic vibrations in molecules and in a lattice, (violin) strings, sound waves, etc.

Planar pendulum

A planar pendulum is an example of harmonic oscillations of a single point-like object undergoing one-dimensional motion.

For small displacements from its equilibrium point (small-angle approximation) the motion of a pendulum can be described by linear equations. A pendulum with one coordinate degree of freedom (a planar pendulum) will act as a simple harmonic oscillator. Classically the equation of motion is used to calculate the position of the pendulum over time,

$$(\ddot{q} + \omega^2 q) = 0, \quad \text{with solution} \quad q(t) = q_0 \cos(\omega t + \phi). \tag{5.1}$$

Here q would actually be (proportional to) the angular displacement from equilibrium, and $\omega = 2\pi\nu$ with ν the oscillation frequency (ω is generally referred to as the angular frequency; this is unrelated to q being an angle in this case).

Atomic vibrations in molecules

Atomic vibrations in molecules is an example of several point-like objects undergoing harmonic motion.

Consider a 3-atomic molecule, as illustrated in figure 5.2, and let $\mathbf{q}_i$ be the fluctuation from the equilibrium position $\mathbf{r}_{0i}$ of particle i,

$$\mathbf{q}_i = (\mathbf{r}_i - \mathbf{r}_{0i}), \quad i = 1, 2, 3.$$

Assume masses $m_1 = m_3 \neq m_2$ and short range interactions, such that only nearest the potential energy between neighbors need be considered. Then the dynamics can be described by the Lagrangian

$$L = T - V, \tag{5.2a}$$

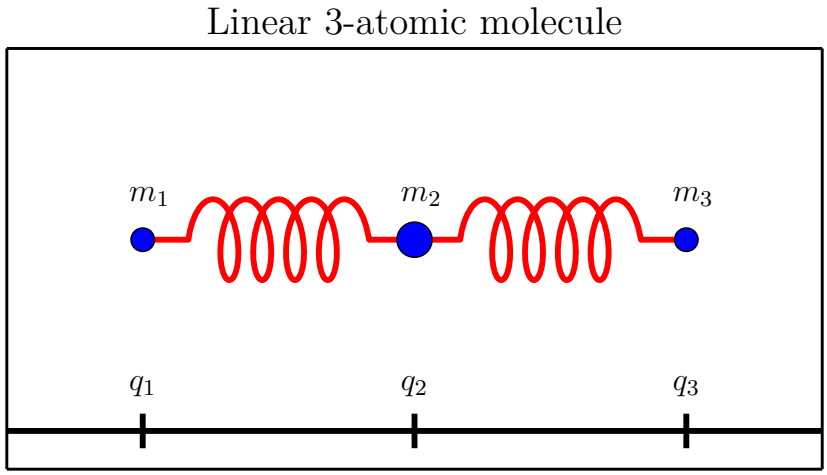


Figure 5.2: Example of a linear 3-atomic molecule, simplified to one-dimensional motion.

with the kinetic energy

$$T = \frac{1}{2}m_1\left(\dot{\mathbf{q}}_1^2 + \dot{\mathbf{q}}_3^2\right) + \frac{1}{2}m_2\dot{\mathbf{q}}_2^2 \tag{5.2b}$$

and the potential energy

$$V = \frac{1}{2}k(\mathbf{q}_1 - \mathbf{q}_2)^2 - \frac{1}{2}k(\mathbf{q}_2 - \mathbf{q}_3)^2. \tag{5.2c}$$

Due to translation symmetry the center of mass

$$\mathbf{Q}_{\rm CM} = [m_1\,(\mathbf{q}_1 + \mathbf{q}_3) + m_2\,\mathbf{q}_2]/M, \tag{5.3}$$

will be unaffected by the internal atomic motion, and will move with constant speed. Here $M = 2m_1 + m_3$ is the total mass. We can choose coordinates such that the center of mass is at rest at the origin, $\mathbf{Q}_{\rm CM} = 0$, and use this to eliminate

$$\mathbf{q}_2 = -(m_1/m_2)(\mathbf{q}_1 + \mathbf{q}_3) = -(m_1/m_2)\mathbf{Q}_a,$$

where we have introduced new coordinates

$$\mathbf{Q}_a = \mathbf{q}_1 + \mathbf{q}_3 \quad \text{and} \quad \mathbf{Q}_s = \mathbf{q}_1 - \mathbf{q}_3. \tag{5.4}$$

Expressed by these variables we find

$$T = \frac{1}{4}m_1\dot{\mathbf{Q}}_s^2 + \frac{1}{4}m_1\left(\frac{M}{m_2}\right)\dot{\mathbf{Q}}_a^2, \qquad V = \frac{1}{4}k\,\mathbf{Q}_s^2 + \frac{1}{4}k\left(\frac{M}{m_2}\right)^2\mathbf{Q}_a^2,$$

such that the Lagrangian (5.2a) can be written as a sum of two independent isotropic three-dimensional systems

$$L = L_s + L_a = \frac{1}{2}m_s\left(\dot{\mathbf{Q}}_s^2 + \omega_s^2\mathbf{Q}_s^2\right) + \frac{1}{2}m_a\left(\dot{\mathbf{Q}}_a^2 + \omega_a^2\mathbf{Q}_a^2\right) \tag{5.5a}$$

$$= \frac{1}{2}\left(\dot{\mathbf{\Theta}}_s^2 + \omega_s^2\mathbf{\Theta}_s^2\right) + \frac{1}{2}\left(\dot{\mathbf{\Theta}}_a^2 + \omega_a^2\mathbf{\Theta}_a^2\right). \tag{5.5b}$$

where $\mathbf{\Theta}_s = \sqrt{m_1/2}\,\mathbf{Q}_s$, $\mathbf{\Theta}_a = \sqrt{m_1 M/(2m_2)}\,\mathbf{Q}_a$, and

$$m_s = \frac{1}{2}m_1, \qquad \omega_s^2 \equiv \frac{k_s}{m_s} = \frac{k}{m_1}, \tag{5.6a}$$

$$m_a = \frac{1}{2}m_1(m_2/M), \qquad \omega_a^2 \equiv \frac{k_a}{m_a} = \frac{kM}{m_1 m_2}. \tag{5.6b}$$

One can transform the above calculations to a more general situation by introducing mass and interaction matrices M_{ij} and K_{ij}, such that the Lagrangian can be written

$$L = \frac{1}{2}\dot{\mathbf{q}}_i M_{ij}\dot{\mathbf{q}}_j - \frac{1}{2}\mathbf{q}_i K_{ij}\mathbf{q}_j, \tag{5.7}$$

where we use the Einstein summation convention (implicit sum over indices that are repeated twice).

$$M_{ij}\ddot{\mathbf{q}}_j + K_{ij}\mathbf{q}_j = 0. \tag{5.8}$$

We make the solution ansatz that $\boldsymbol{q}_i \propto \boldsymbol{q}_i^{(0)}\mathrm{e}^{-\mathrm{i}\omega t}$ (or the real or imaginary part of this) to transform the equations of motion (5.7) to an algebraic equation

$$K_{ij}\,\mathbf{q}_j^{(0)} = \omega^2 M_{ij}\,\mathbf{q}_j^{(0)}, \tag{5.9}$$

which is a (generalized) eigenvalue problem, with ω^2 being the eigenvalues. They are determined by the algebraic determinant conditions

$$\left|\mathbb{K} - \omega^2\mathbb{M}\right| = 0. \tag{5.10a}$$

where $\mathbb{K}$ is the symmetric matrix with elements K_{ij} and $\mathbb{M}$ the symmetric matrix with elements M_{ij}. The mass matrix $\mathbb{M}$ should be positive definite

(and is usually diagonal), so we may define its inverse square root $\mathbb{M}^{-1/2}$, and transform (5.10a) to a normal eigenvalue condition

$$\left|\mathbb{M}^{-1/2}\,\mathbb{K}\,\mathbb{M}^{-1/2} - \omega^2\mathbb{1}\right| = 0, \tag{5.10b}$$

Compared with T and V in equations(5.2b) and (5.2c) we find

$$\mathbb{M} = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_1 \end{bmatrix}, \qquad \mathbb{K} = \begin{bmatrix} k & -k & 0 \\ -k & 2k & -k \\ 0 & -k & k \end{bmatrix}, \tag{5.11}$$

and equation (5.10a) becomes explicitly,

$$\mathrm{Det} = \begin{vmatrix} k-\omega^2 m_1 & -k & 0 \\ -k & 2k-\omega^2 m_2 & -k \\ 0 & -k & k-\omega^2 m_1 \end{vmatrix} = 0.$$

An expansion of the determinant shows that

$$\mathrm{Det} = -\omega^2\left(m_1\omega^2 - k\right)\left[m_1 m_2\omega^2 - (2m_1 + m_1)k\right],$$

in agreement with the results (5.6).

Atomic vibrations in a lattice

An infinite atomic (or molecular) lattice is an example of harmonic oscillations of an infinite set of point-like objects. It reveals a continuous infinity of oscillation frequencies, distributed over a finite range.

From the point of view of the discussion above, a lattice can be considered as the infinite size limit of a very large molecule. This limit will usually have lattice translations symmetry, which helps to solve the problem with the use of the Fourier transformation technique. Since the system consists of an infinite number of harmonic oscillators, it will have infinitely many oscillation frequencies, all of them within in a bounded range.

Strings

A (mathematically idealized) string of finite length is an example of harmonic oscillations of a one-dimensional continuous object. It reveals an infinity of discrete oscillation frequencies, increasing to infinity.

Consider a thin homogeneous string in the plane, stretched out between two fixed points $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$. We may orient coordinates such that these points lie on the x-axis, $\boldsymbol{x}_0 = (x_0, 0)$ and $\boldsymbol{x}_1 = (x_1, 0)$, and further label the

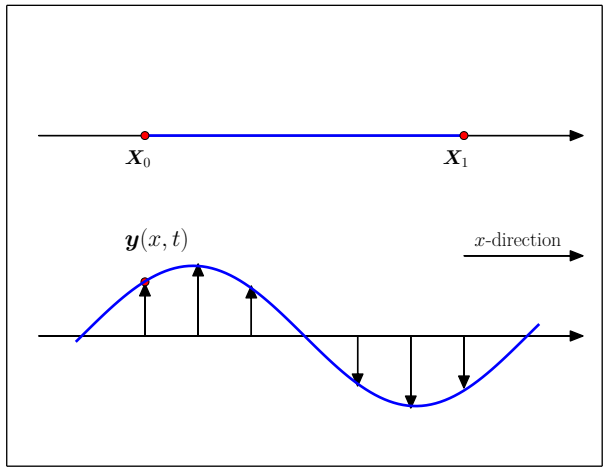


Figure 5.3: *A string in the xy-plane.*

points of the string by a parameter λ such that $x(\lambda) = x_0 + (x_1 - x_0)\lambda$, $y(\lambda) = 0$ when the string is at rest, with $0 \leq \lambda \leq 1$. The string may now be set in motion, with mainly transverse displacement $y(\lambda)$ of its points.

The equation of motion for this displacement is

$$\rho\, \ddot{y}(t,\lambda) = \sigma y''(t,\lambda), \quad y(t,0) = y(t,1) = 0, \tag{5.12}$$

where ρ is the mass density per unit of length (as measured by λ) and σ is a tension force. We can expand the displacement in a trigonometric series

$$y(t,\lambda) = \sum_{n=1}^{\infty} y_n(t)\, \sin(n\pi\lambda),$$

which inserted into (5.12) gives an infinite discrete sequence of harmonic oscillator equations

$$\ddot{y}_n + \omega_n^2\, y_n = 0, \qquad n = 1, 2, \ldots, \tag{5.13}$$

where $\omega_n^2 = (n\pi)^2(\sigma/\rho)$. Here $c_s \equiv \sqrt{\sigma/\rho}$ is the wave propagation speed along the string. Note that all frequencies ω_n is an integer multiple of the basic frequency ω_1.

Sound waves

The (mathematically idealized) model for sound waves in infinite space is an example of harmonic oscillations in three-dimensional continuous fluid. It reveals an continuous infinity of oscillation frequences, increasing to infinity.

Similar to string equation of motion (5.12), sound equation of motion in a fluid becomes

$$\frac{\partial^2 P}{\partial t^2} = \frac{K}{\rho}\frac{\partial^2 P}{\partial x^2} \tag{5.14}$$

where $K = \rho_0 \left(\partial P/\partial \rho\right)_{\text{adiabatic}}$ is the bulk modulus of the fluid [48].

In contrast to the string equation, this equation describes how the pressure in the fluid varies with space and time. Hence it is not directly related to any displacements in space; it is a *scalar* wave equation. (The wave will propagate in space, of course.)

A solution to equation (5.14) is,

$$P = P_m \sin\left(kx - \omega t\right). \tag{5.15}$$

By deriving equation (5.15) we get,

$$\frac{\partial^2 P}{\partial t^2} = -P_m \omega^2 \sin\left(km - \omega t\right). \tag{5.16}$$

And,

$$\frac{\partial^2 P}{\partial x^2} = -P_m k^2 \sin\left(km - \omega t\right). \tag{5.17}$$

So the speed c_s of sound wave is

$$c_s = \sqrt{\frac{K}{\rho}}. \tag{5.18}$$

In the atmosphere, the speed of sound waves are

$$c_s = \sqrt{\frac{\gamma P_{\text{eq}}}{\rho}}, \tag{5.19}$$

where P_{eq} is the equilibrium atmospheric pressure.

5.2 Thermal and quantum fluctuations

Everything in this universe undergoes fluctuations. In classical physics this requires a finite temperature, leading to thermal fluctuations, but in quantum physics there are fluctuations even at zero absolute temperature (zero-point fluctuations). As the temperature increases from zero, the quantum

fluctuations gradually term turn into thermal ones. Hence ought to be possible to describe them in a common formalism, treating them in a similar fashion.

Ny assuming the flutuations to be small around an equilibrium configuration, they can be treated as fluctuations of a collection of harmonic oscillators with various frequencies.

5.2.1 Classical equipartition principle

Thermal fluctuations of classical particles at finite temperature T can in the first approximation be described by the equipartition principle. The idea is that the total energy of the system on the average will be shared equally among all dynamical degrees of freedom (dof) of the system, whether they correspond translational, rotational or vibrational motion,

$$E_{\text{dof}} = \frac{1}{2} k_B T \tag{5.20}$$

where $k_B = 1.381 \times 10^{-23}\ \text{m}^2\,\text{kg}\,\text{s}^{-2}\,\text{K}^{-1}$ is the Boltzmann constant. Hence it becomes a question of counting the (effective) number N_e of degrees of freedom.

- Each freely moving point particle in three-dimensional space contribute $N_e = 3$, one for each velocity direction (the position degree of freedom does not count when the potential is zero).

- Each independent mode of harmonic oscillation contribute $N_e = 2$, one from velocity and one from position.

- The rotation of linar (i.e two-atomic) molecules contributes $N_e = 2$, from the motion of the motion of molecule orientation (rotation around the symmetry axis of the molecule does not count).

- The rotation of larger molecules contributes $N_e = 3$, since rotation around all three axes of the molecule counts.

5.2.2 Quantum zero-point fluctuation

For quantum mechanical zero-point fluctuations, consider the ground state fluctuations in position a haromic oscillator,

$$\text{Var}(\hat{q}) = \langle 0|\hat{q}^2|0\rangle - \langle 0|\hat{q}|0\rangle^2. \tag{5.21}$$

By writing $\hat{q}$ in terms of annihilation and creation operators [see eqs. (2.32)]

$$\hat{q} = \sqrt{\frac{\hbar}{2m\omega}} \left(a + a^\dagger \right), \tag{5.22}$$

we find $\langle 0|\hat{q}|0\rangle = 0$, and

$$\mathrm{Var}(\hat{q}) = \langle 0|\hat{q}^2|0\rangle = \frac{\hbar}{2m\omega}\langle 0|aa^\dagger|0\rangle = \frac{\hbar}{2m\omega}, \tag{5.23}$$

since $a|0\rangle = 0$, $\langle 0|a^\dagger = 0$, and $aa^\dagger = a^\dagger a + 1$. By comparison with the classical equipartition principle one finds that quantum fluctuations correspond to an effective temperature T_{qu} such that

$$k_B T_{\mathrm{qu}} = \hbar\omega. \tag{5.24}$$

5.3 Classical unforced and undamped oscillator

In this and the following sections we will first analyse a classical unforced and undamped oscillator (section 5.3), next a classical forced and damped oscillator (section 5.4), then Langevin equations (section 5.5) and finally the linear harmonic chain (section 5.6).

The harmonic oscillator Hamiltonian for one-dimensional motion is

$$H = \frac{1}{2}p^2 + \frac{1}{2}q^2. \tag{5.25}$$

Here the coordinate q describe the (small) deviation from a equilibrium position, and the momentum $p = mv$ the corresponding velocity. We have choosen units for mass and time such that particle mass $m = 1$ and angular oscillation frequency $\omega = 1$.

Accordingly Hamilton's equations of motion, equations (2.12) for this system becomes

$$\dot{q} = \frac{\partial H}{\partial p} = p, \tag{5.26a}$$

$$\dot{p} = -\frac{\partial H}{\partial q} = -q, \tag{5.26b}$$

where $\dot{}$ denotes time derivative. Equation (5.26a) just reproduces the (defined) relation between momentum and velocity. Equation (5.26b) shows that the system introduces a linear restoring force which opposes the displacement. This is necessary to keep stability of the system.

In matrix form both equations (5.26b) can be written as,

$$\left[\frac{d}{dt}+\mathcal{D}\right]\begin{pmatrix}q(t)\\p(t)\end{pmatrix}=0,\quad \mathcal{D}=\begin{pmatrix}0&-1\\1&0\end{pmatrix}. \tag{5.27}$$

Note that we have use a dimensionless time τ, and dimensionless coordinates q, p.

To solve Hamilton's equations, we rewrite (5.27) as

$$e^{-t\mathcal{D}}\frac{d}{dt}e^{t\mathcal{D}}\begin{pmatrix}q(t)\\p(t)\end{pmatrix}=0,\quad \text{or}\quad \frac{d}{dt}e^{t\mathcal{D}}\begin{pmatrix}q(t)\\p(t)\end{pmatrix}=0. \tag{5.28}$$

Integrated from 0 to t this gives [1],

$$e^{t\mathcal{D}}\begin{pmatrix}q(t)\\p(t)\end{pmatrix}-\begin{pmatrix}q_0\\p_0\end{pmatrix}=0,$$

or, (5.29)

$$\begin{pmatrix}q(t)\\p(t)\end{pmatrix}=e^{-t\mathcal{D}}\begin{pmatrix}q_0\\p_0\end{pmatrix}=\begin{pmatrix}\cos(t)&\sin(t)\\-\sin(t)&\cos(t)\end{pmatrix}\begin{pmatrix}q_0\\p_0\end{pmatrix}.$$

We can use these solutions for more general problems; 1) by rescaling the position and momentum coordinates, and 2) by translating the position coordinate.

Consider the Hamiltonian

$$H=\frac{1}{2m}P^2+\frac{1}{2}KQ^2\equiv\frac{1}{2m}P^2+\frac{1}{2}m\omega^2Q^2. \tag{5.30}$$

Here $\omega^2=K/m$. By introducing new coordinates [2], $q=(m\omega)^{-1/2}Q$ and $p=(m\omega)^{1/2}P$, we find

$$H=\frac{1}{2}\omega(p^2+q^2). \tag{5.31}$$

This is the same Hamiltonian as in (5.25), with $\tau=\omega t$.Hence, the solution becomes

$$\begin{aligned}\begin{pmatrix}Q(t)\\P(t)\end{pmatrix}=&V\begin{pmatrix}\cos(\omega t)&\sin(\omega t)\\-\sin(\omega t)&\cos(\omega t)\end{pmatrix}V\begin{pmatrix}Q_0\\P_0\end{pmatrix}\\=&\begin{pmatrix}\cos(\omega t)&m\omega\sin(\omega t)\\-(m\omega)^{-1}\sin(\omega t)&\cos(\omega t)\end{pmatrix}\begin{pmatrix}Q_0\\P_0\end{pmatrix}.\end{aligned} \tag{5.32}$$

[1] We can verify directly that this expression satisfies (5.26*b*) or, we may note that $\mathcal{D}^2=-\mathbb{1}$, which implies that,

$$e^{-t\mathcal{D}}=\sum_{n=0}^{\infty}\frac{1}{n!}(-t\mathcal{D})^n=\sum_{m=0}^{\infty}\frac{(-\mathbb{1})^m}{2m!}t^{2m}-\sum_{m=0}^{\infty}\frac{(-1)^m}{(2m+1)!}t^{2m+1}\mathcal{D}=\cos(t)\,\mathbb{1}-\sin(t)\,\mathcal{D}$$

[2] Defined such that the Poisson bracket is maintained, {p,q}={P,Q}.

Where

$$V = \begin{pmatrix} (m\omega)^{1/2} & 0 \\ 0 & (m\omega)^{-1/2} \end{pmatrix},$$

and

$$U = \begin{pmatrix} \cos(\omega t) & m\omega \sin(\omega t) \\ -(m\omega)^{-1} \sin(\omega t) & \cos(\omega t) \end{pmatrix}.$$

We generalize the Hamiltonian once more,

$$\begin{aligned} H &= \frac{1}{2m} P^2 + \frac{1}{2} m\omega^2 Q^2 - FQ \\ &= \frac{1}{2m} P^2 + \frac{1}{2} m\omega^2 \left[(Q - \bar{Q})^2 - \bar{Q}^2 \right] - FQ. \end{aligned} \tag{5.33}$$

Here F is an external constant force, and $\bar{Q} = F/(m\omega^2)$. The constant $-(1/2)m\omega^2\bar{Q}^2$ is a shift in energy which does not contribute to the equations of motion; hence we find

$$\begin{pmatrix} Q(t) - \bar{Q} \\ P(t) \end{pmatrix} = U \begin{pmatrix} Q_0 - \bar{Q} \\ P_0 \end{pmatrix}, \tag{5.34}$$

or

$$\begin{pmatrix} Q(t) \\ P(t) \end{pmatrix} = U \begin{pmatrix} Q_0 \\ P_0 \end{pmatrix} + \begin{pmatrix} 1 - \cos(\omega t) \\ (m\omega)^{-1} \sin(\omega t) \end{pmatrix} \frac{F}{(m\omega^2)}. \tag{5.35}$$

5.4 Classical forced and damped oscillator

5.4.1 Forced oscillator

Next modify the Hamiltonian by addition of a time dependent external force,

$$H \longrightarrow \frac{1}{2} p^2 + \frac{1}{2} q^2 - f(t)\, q. \tag{5.36}$$

This changes the Hamilton's equations to

$$\dot{q} = p, \qquad \dot{p} = -q + f(t). \tag{5.37}$$

In matrix form we now get

$$e^{-t\mathcal{D}} \frac{d}{dt} e^{t\mathcal{D}} \begin{pmatrix} q(t) \\ p(t) \end{pmatrix} = \begin{pmatrix} 0 \\ f(t) \end{pmatrix} \tag{5.38}$$

or,

$$\frac{d}{dt} \left[e^{t\mathcal{D}} \begin{pmatrix} q(t) \\ p(t) \end{pmatrix} \right] = e^{t\mathcal{D}} \begin{pmatrix} 0 \\ f(t) \end{pmatrix}.$$

Integrated for 0 to t

$$e^{t\mathcal{D}}\begin{pmatrix} q\left(t\right) \\ p\left(t\right) \end{pmatrix} - \begin{pmatrix} q_0 \\ p_0 \end{pmatrix} = \int_0^t du\; e^{u\mathcal{D}}\begin{pmatrix} 0 \\ f\left(u\right) \end{pmatrix}.$$

I.e.

$$\begin{pmatrix} q\left(t\right) \\ p\left(t\right) \end{pmatrix} = e^{-t\mathcal{D}}\begin{pmatrix} q_0 \\ p_0 \end{pmatrix} + \int_0^t du\; e^{-(t-u)\mathcal{D}}\begin{pmatrix} 0 \\ f\left(u\right) \end{pmatrix}. \tag{5.39}$$

To proceed from here we need to know more about the function $f\left(u\right)$. As a simple check, assume it is constant. Then, by use of $\int_0^t du \cos(t-u) = \sin(t)$, $\int_0^t du \sin(t-u) = 1-\cos(t)$ we find

$$\int_0^t du\; e^{-(t-u)\mathcal{D}}\begin{pmatrix} 0 \\ f \end{pmatrix} = \begin{pmatrix} 1-\cos(t) \\ \sin(t) \end{pmatrix} f. \tag{5.40}$$

With this result inserted into (5.74a) we see that it agrees with eq.(5.35) when $m = \omega = 1$.

5.4.2 Damped oscillator

We next introduce a (small) damping term. Then the system no longer can be described by a Hamiltonian. The Hamilton's equations (5.37) changes to the more general dynamical equation

$$\dot{q} = p, \qquad \dot{p} = -q + f\left(t\right) - \gamma p. \tag{5.41}$$

This can still be formulated in matrix form, like in eq.(5.38)

$$e^{-t\mathcal{D}}\frac{d}{dt}e^{t\mathcal{D}}\begin{pmatrix} q\left(t\right) \\ p\left(t\right) \end{pmatrix} = \begin{pmatrix} 0 \\ f\left(t\right) \end{pmatrix}, \tag{5.42}$$

but now with a dynamical matrix

$$\mathcal{D} = \begin{pmatrix} 0 & -1 \\ 1 & \gamma \end{pmatrix}. \tag{5.43}$$

This matrix has eigenvalues $\lambda_\pm = \frac{1}{2}\gamma \pm i\sqrt{1-\frac{\gamma^2}{4}}$, satisfying $\lambda_+ + \lambda_- = \gamma$ and $\lambda_+\lambda_- = 1$. It can be decomposed as

$$\mathcal{D} = R\begin{pmatrix} \lambda_+ & 0 \\ 0 & \lambda_- \end{pmatrix}R^{-1}, \tag{5.44}$$

where R is a matrix of right eigenvectors,

$$R = \begin{pmatrix} 1 & -\lambda_+ \\ -\lambda_+ & 1 \end{pmatrix}, \qquad R^{-1} = \frac{1}{1-\lambda_+^2}\begin{pmatrix} 1 & \lambda_+ \\ \lambda_+ & 1 \end{pmatrix} \equiv L. \tag{5.45}$$

Hence, any function of $\mathcal{D}$ can be decomposed as

$$f(\mathcal{D}) = R\begin{pmatrix} f(\lambda_+) & 0 \\ 0 & f(\lambda_-) \end{pmatrix} R^{-1}. \tag{5.46}$$

In particular we may write, for $\gamma^2 \leq 4$,

$$e^{-(t-u)\mathcal{D}} = e^{-\frac{1}{2}(t-u)\gamma} R \begin{pmatrix} e^{-i(t-u)w_s} & 0 \\ o & e^{i(t-u)\omega_s} \end{pmatrix} R^{-1} \tag{5.47}$$

where $\omega_s = \sqrt{1-\gamma^2/4}$. Note that e^{-tD} decreases exponentially with time like $e^{-\gamma t/2}$.

5.4.3 Forced and damped oscillator

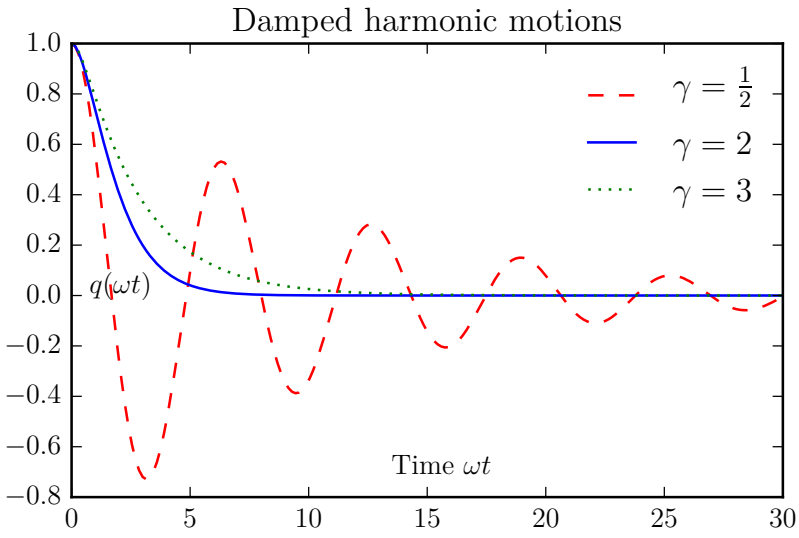


Figure 5.4: *The possible time evolutions of some damped harmonic oscillators. Note that a critically damped oscillator, $\gamma = 2$, will have the fastest decay to equilbrium. Underdamped oscillators (here with $\gamma = \frac{1}{2}$) decay in an oscillatory manner.*

The general solution can be written as

$$\begin{pmatrix} q\,(t) \\ p\,(t) \end{pmatrix} = \overbrace{R\,e^{-tD}\,L}^{e^{-t\mathcal{D}}} \begin{pmatrix} q_0 \\ p_0 \end{pmatrix} + \int_0^t du\ \overbrace{R\ e^{-(t-u)D}\,L}^{e^{-(t-u)\mathcal{D}}} \begin{pmatrix} 0 \\ f\,(u) \end{pmatrix}. \tag{5.48}$$

Here the first term of the right side of eq.(5.48) has a common factor

$$e^{-\frac{1}{2}\gamma t} R\ e^{\mp i\omega_s t} L\ \begin{pmatrix} q_0 \\ p_0 \end{pmatrix}, \tag{5.49}$$

which means that the dependence on the initial values q_0, p_0 is attenuated exponentially when t becomes large, proportional to $e^{-\gamma t/2}$. One may say that the system has acquired short time memory. This makes it simpler to find the large-time solution. Assume an oscillatory force, $f(u) = f_0 \cos\omega t = \mathrm{Re}\left(f_0 \mathrm{e}^{i\omega t}\right)$, and a solution which oscillates with the same frequency,

$$\begin{pmatrix} q(t) \\ p(t) \end{pmatrix} = \mathrm{Re}\left[\begin{pmatrix} \overline{q} \\ \overline{p} \end{pmatrix} \mathrm{e}^{i\omega t} \right]. \tag{5.50}$$

Here $\bar{q}$ and $\bar{p}$ are complex constants which depend on the driving force (and other parameters). Since taking the real part of an expression is a linear operation, we have

$$\begin{pmatrix} \dot{q}(t) \\ \dot{p}(t) \end{pmatrix} = \mathrm{Re}\left[i\omega \begin{pmatrix} \overline{q} \\ \overline{p} \end{pmatrix} e^{i\omega t} \right]. \tag{5.51}$$

Since $\mathrm{e}^{-i\omega t}$ is never zero, insertion into equation (5.41) leads to the condition

$$\begin{pmatrix} i\omega & -1 \\ 1 & i\omega + \gamma \end{pmatrix} \begin{pmatrix} \overline{q} \\ \overline{p} \end{pmatrix} = \begin{pmatrix} 0 \\ f_0 \end{pmatrix}, \tag{5.52}$$

with solution

$$\begin{aligned} \begin{pmatrix} \overline{q} \\ \overline{p} \end{pmatrix} =& \frac{1 - \omega^2 - i\gamma\omega}{(1-\omega^2)^2 + (\gamma\omega)^2} \begin{pmatrix} i\omega + \gamma & 1 \\ -1 & i\omega \end{pmatrix} \begin{pmatrix} 0 \\ f_0 \end{pmatrix} \\ =& \frac{1 - \omega^2 - i\gamma\omega}{(1-\omega^2)^2 + (\gamma\omega)^2} \begin{pmatrix} 1 \\ i\omega \end{pmatrix} f_0 \equiv R(\omega)\ e^{-i\phi} \begin{pmatrix} 1 \\ e^{i(\pi/2)}\omega \end{pmatrix} f_0. \end{aligned} \tag{5.53}$$

Here

$$R(\omega) = \frac{1}{\sqrt{(1-\omega^2)^2 + (\gamma\omega)^2}}, \qquad \tan\phi = \frac{\gamma\omega}{1-\omega^2}. \tag{5.54}$$

To proceed to the solution for a general driving force, the time integral in equation (5.48) must be done numerically. Or, due to linearity, one may superpose oscillatory solutions of the type found above (but the latter will also require numerical work to comprehend).

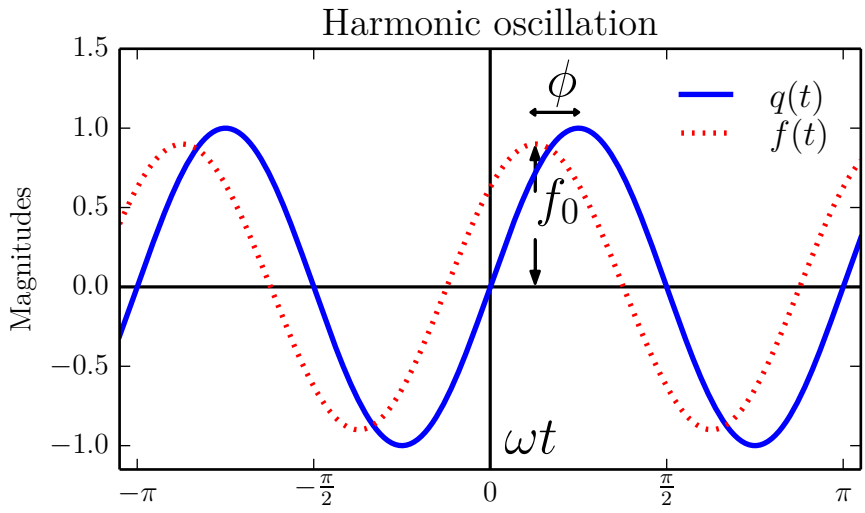


Figure 5.5: This figure show the long time behavior of a damped oscillator, driven with an oscillatory force of constant amplitude. The two quantities of main interest are (i) the ratio f_0 between the source and force amplitudes, and (ii) the phase difference ϕ between the two quantities.

5.4.4 Forced and damped oscillators

The explicit analysis of the previous section can be carried over, in symbolic form, to the general multicomponent case. The equation of motion can be writtem

$$\underbrace{e^{-u\mathcal{D}}\frac{d}{du}e^{u\mathcal{D}}}_{\left(\frac{d}{du}+\mathcal{D}\right)} z\left(u\right) = f\left(u\right). \tag{5.55}$$

Here z is a multicomponent vector of all positions and momenta, $\mathcal{D}$ a general dynamical matrix, and $f(u)$ a multicomponent vector of (generalized) time-dependent forces. We rewrite this equation as

$$\frac{d}{du}e^{u\mathcal{D}}z(u) = e^{u\mathcal{D}}f(u), \tag{5.56}$$

and integrate over u from $u = 0$ to t, to find

$$e^{t\mathcal{D}}z(t) - z(0) = \int_0^t du\, e^{u\mathcal{D}}f(u). \tag{5.57}$$

I.e., the general formal solution is

$$z(t) = e^{-t\mathcal{D}} \left[z(0) + \int_0^t du\, e^{u\mathcal{D}} f(u) \right]. \tag{5.58}$$

This form is a useful starting point for numerical work. We only have to find a spectral representation of the dynamical matrix $\mathcal{D}$. From the point of view of numerical routines, $\mathcal{D}$ is a general $2N \times 2N$-dimensional real matrix, with complex eigenvalues and eigenvectors.

To proceed a bit further analytically, assume that $f(u) = \cos \omega u\, f_0$, We must do the integral

$$\begin{aligned}
&\frac{1}{2} \int_0^t du\, e^{u\mathcal{D}} \left(\mathrm{e}^{i\omega u} + \mathrm{e}^{-i\omega u} \right) \\
&= \frac{1}{2} \int_0^t du\, \left[\mathrm{e}^{u(\mathcal{D}+i\omega)} + \mathrm{e}^{u(\mathcal{D}-i\omega)} \right] \\
&= \frac{1}{2} \left[\frac{1}{\mathcal{D}+i\omega}\, \mathrm{e}^{u(\mathcal{D}+i\omega)} + \frac{1}{\mathcal{D}-i\omega}\, \mathrm{e}^{u(\mathcal{D}-i\omega)} \right]_{u=0}^{u=t} \\
&= \frac{1}{2} \frac{(\mathcal{D}-i\omega) \left[e^{t(\mathcal{D}+i\omega)} - 1 \right] + (\mathcal{D}+i\omega) \left[e^{t(\mathcal{D}-i\omega)} - 1 \right]}{(\mathcal{D}+i\omega)\,(\mathcal{D}-i\omega)} \\
&= \frac{1}{2} \frac{\mathcal{D}\, \mathrm{e}^{t\mathcal{D}} (\mathrm{e}^{i\omega t} + \mathrm{e}^{-i\omega t}) - i\omega \mathrm{e}^{t\mathcal{D}} (\mathrm{e}^{i\omega t} - \mathrm{e}^{-i\omega t})}{(\mathcal{D}^2 + \omega^2)} - \frac{(\mathcal{D} - i\omega + \mathcal{D} + i\omega)}{(\mathcal{D}^2 + \omega^2)} \\
&= \mathrm{e}^{t\mathcal{D}} \left[\frac{\mathcal{D}\, \cos(\omega t) + \omega\, \sin(\omega t)}{(\mathcal{D}^2 + \omega^2)} \right] - \frac{\mathcal{D}}{(\mathcal{D}^2 + \omega^2)}.
\end{aligned}$$

We have made use the fact that $\mathcal{D}$ is the only matrix in the problem. Since it commutes with itself, it can formally be treated as an ordinary number. But one must be aware that quantities must be understood as matrix products, inverses, exponentials. The above computation can be summarized as

$$\begin{aligned}
\mathrm{e}^{-t\mathcal{D}} \int_0^t du\, e^{u\mathcal{D}} \cos \omega u &= \frac{\mathcal{D}\, \cos \omega t + \omega\, \sin \omega t - \mathcal{D} \mathrm{e}^{-t\mathcal{D}}}{(\mathcal{D}^2 + \omega^2)} \\
&= R \left[\frac{D\, \cos(\omega t) + \omega \sin(\omega t) - D\, e^{-tD}}{(D^2 + \omega^2)} \right] L
\end{aligned} \tag{5.59}$$

where we have performed a spectral decomposition $\mathcal{D} = R\, D\, L$, with D the diagonal matrix of eigenvalues of $\mathcal{D}$, and R and L the corresponding matrices of right and left eigenvectors, properly normalized so that $R\, L = L\, R = \mathbb{1}$.

5.5 Langevin equations

The equations to be solved are typically of the form

$$\dot{q}_m = \frac{\partial H}{\partial p_m} = M^{-1}_{mn}\, p_m, \tag{5.60}$$

$$\dot{p}_m = -\frac{\partial H}{\partial q_m} - \Gamma_{mn}\, p_n + f_m, \tag{5.61}$$

where f_m are stochastic forces (usually with a gaussian distribution) and M is a symmetric positive definite matrix and Γ is a symmetric non-negative matrix (both usually diagonal). The *summation convention* that repeated indices are summed over is used. In the first approximation one may assume the equations to be linear, i.e. that $\partial H/\partial q_m = K_{mn} q_n$. The equation to be solved becomes

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} \mathbf{q}(t) \\ \mathbf{p}(t) \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{M}^{-1} \\ -\mathbf{K} & -\mathbf{\Gamma} \end{pmatrix}\begin{pmatrix} \mathbf{q}(t) \\ \mathbf{p}(t) \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \mathbf{f}(t) \end{pmatrix}, \tag{5.62}$$

where boldface denotes $N \times N$ matrices and N-component vectors. This can be rewritten in compact form as

$$\frac{\mathrm{d}}{\mathrm{d}t}\, \mathrm{e}^{t\mathcal{D}}\, \mathbf{y}(t) = \mathrm{e}^{t\mathcal{D}}\, \mathbf{x}(t), \tag{5.63}$$

where boldface now denotes $2N$-component vectors and $2N \times 2N$ matrices,

$$\mathbf{y}(t) = \begin{pmatrix} \mathbf{q}(t) \\ \mathbf{p}(t) \end{pmatrix}, \quad \mathbf{x}(t) = \begin{pmatrix} \mathbf{0} \\ \mathbf{f}(t) \end{pmatrix}, \quad \mathcal{D} = \begin{pmatrix} \mathbf{0} & -\mathbf{M}^{-1} \\ \mathbf{K} & \mathbf{\Gamma} \end{pmatrix}. \tag{5.64}$$

Choose a timestep h and integrate equation (5.68) from $t = kh$ to $t = (k+1)h$. One obtains

$$\mathbf{y}((k+1)h) = \mathrm{e}^{-h\mathcal{D}}\, \mathbf{y}(kh) + \mathrm{e}^{-(k+1)h\mathcal{D}} \int_{kh}^{(k+1)h} \mathrm{e}^{t\mathcal{D}} \mathbf{x}(t)\, \mathrm{d}t. \tag{5.65}$$

Define

$$\mathbf{y}_k \equiv \mathbf{y}(kh), \quad \mathcal{G} \equiv \mathrm{e}^{-h\mathcal{D}}, \quad \text{and} \quad \mathbf{x}_k \equiv \mathrm{e}^{-(k+1)h\mathcal{D}} \int_{kh}^{(k+1)h} \mathrm{e}^{t\mathcal{D}} \mathbf{x}(t)\, \mathrm{d}t,$$

to rewrite (5.65) as

$$\mathbf{y}_{k+1} = \mathcal{G}\mathbf{y}_k + \mathbf{x}_k. \tag{5.66}$$

This equation is an exact consequence of (5.68). Here the dependence on the timestep h is not indicated explicitly.
Then the eq. (5.82) will take the form,

$$\mathbb{C}^1_{mn} = K \sum_{kl} \frac{R_{mk} L_{k_2} \; R_{nl} \; L_{l_2}}{(\lambda_k + \lambda_l)}. \tag{5.67}$$

5.5.1 Langevin Description of Thermalization

Consider a set of linear stochastic differential equations

$$m_j \ddot{x}_j + K_{jk} x_k = \delta_{j1} \left(f_1 - m_1 \gamma \, \dot{x}_1 \right), \tag{5.68}$$

where all m_j are positive, and K_{jk} is a positive definite symmetric matrix. We introduce $y_j = \sqrt{m_j} x_j$, i.e. $x_j = y_j / \sqrt{m_j}$, and divide equation (5.68) by $\sqrt{m_j}$. This gives

$$\ddot{y}_j + \mathcal{K}_{jk} y_k = \delta_{j1} \left(g_1 - \gamma \, \dot{y}_1 \right), \tag{5.69}$$

where $\mathcal{K}_{jk} = K_{jk} / \sqrt{m_j \, m_k}$ is also a positive definite symmetric matrix, and $g_1 = f_1 / \sqrt{m_1}$. We have used that $\delta_{j1} / \sqrt{m_j} = \delta_{j1} / \sqrt{m_1}$. $\mathcal{K}_{jk}$ can be diagonalized by a real orthogonal transform,

$$\mathcal{K}_{jk} = \sum_r O_{rj} \, \omega_r^2 \, O_{rk}, \tag{5.70}$$

where $O_{rj} \, O_{rk} = \delta_{jk}$. We define $u_r \equiv O_{rk} \, y_k$ to get

$$O_{rj} \left(\ddot{u}_r + \omega_r^2 u_r \right) = \delta_{j1} \left(g_1 - \gamma \, \dot{y}_1 \right). \tag{5.71}$$

5.5.2 Oscillator influenced by a random force

$$\left[\frac{d}{dt} + \mathcal{D} \right] \begin{pmatrix} q\,(t) \\ p\,(t) \end{pmatrix} = \begin{pmatrix} 0 \\ f\,(t) \end{pmatrix} \tag{5.72}$$

$$\begin{pmatrix} q\,(t) \\ p\,(t) \end{pmatrix} = e^{-t\mathcal{D}} \begin{pmatrix} q_0 \\ p_0 \end{pmatrix} + \int_0^t du \; e^{-(t-u)\mathcal{D}} \begin{pmatrix} 0 \\ f\,(u) \end{pmatrix}. \tag{5.73}$$

Now we asume that $f(u)$ is a Gaussian distributed random function. It is complete charactraized by its mean value $\langle f(u) \rangle = 0$ and fluctuations $\langle f(u) f(v) \rangle = K \delta u - v$. We want to compute the probability distribution of q, p at time t. As a sum (integral) of Gaussian distributed random variables,

it will also be a Gaussian distributed random variable. Hence we need to compute,

$$\begin{pmatrix} q\left(t\right) \\ p\left(t\right) \end{pmatrix} = e^{-t\mathcal{D}} \begin{pmatrix} q_0 \\ p_0 \end{pmatrix} + \int_0^t du\ e^{-(t-u)\mathcal{D}} \begin{pmatrix} 0 \\ f\left(u\right) \end{pmatrix} \tag{5.74a}$$

$$= e^{-t\mathcal{D}} \begin{pmatrix} q_0 \\ p_0 \end{pmatrix}. \tag{5.74b}$$

And,

$$\mathbb{C} \equiv \begin{pmatrix} \langle q\left(t_1\right) q\left(t_2\right) \rangle & \langle q\left(t_1\right) p\left(t_2\right) \rangle \\ \langle p\left(t_1\right) q\left(t_2\right) \rangle & \langle p\left(t_1\right) p\left(t_2\right) \rangle \end{pmatrix} \tag{5.75}$$

$$\begin{aligned} \left\langle \begin{pmatrix} q\left(t_1\right) \\ p\left(t_1\right) \end{pmatrix} \otimes \begin{pmatrix} q\left(t_2\right) \\ p\left(t_2\right) \end{pmatrix} \right\rangle &= e^{-t_1\mathcal{D}} \begin{pmatrix} q_0 \\ p_0 \end{pmatrix} \otimes e^{-t_2\mathcal{D}} \begin{pmatrix} q_0 \\ p_0 \end{pmatrix} \\ &+ e^{-t_1\mathcal{D}} \begin{pmatrix} q_0 \\ p_0 \end{pmatrix} \otimes \int_0^{t_2} du\ e^{-(t_2-u)\mathcal{D}} \begin{pmatrix} 0 \\ \langle f(u) \rangle \end{pmatrix} \\ &+ \int_0^{t_1} du\ e^{-(t_1-u)\mathcal{D}} \begin{pmatrix} 0 \\ \langle f(u) \rangle \end{pmatrix} \otimes e^{-t_2\mathcal{D}} \begin{pmatrix} q_0 \\ p_0 \end{pmatrix} \\ &+ \int_0^{t_1} du_1 \int_0^{t_2} du_2\ e^{-(t_1-u_1)\mathcal{D}} \left\langle \begin{pmatrix} 0 \\ f(u_1) \end{pmatrix} \otimes e^{-(t_2-u_2)\mathcal{D}} \begin{pmatrix} 0 \\ f(u_2) \end{pmatrix} \right\rangle, \end{aligned} \tag{5.76}$$

let $\varepsilon(t) = e^{-(t)\mathcal{D}}$, the last term will be,

$$\int_0^{t_1} du_1 \int_0^{t_2} du_2\ \varepsilon_{n_2}(t_1 - u_1)\varepsilon_{m_2}(t_2 - u_2)\ f(u_1)\, f(u_2). \tag{5.77}$$

We may decompose $\mathbb{C} = \mathbb{C}^{(0)} + \mathbb{C}^{(1)}, where,$

$$C^{(0)}_{mn} = \varepsilon_{mp}(t_1)\varepsilon_{np}(t_2) z^{(0)}_p z^{(0)}_q \tag{5.78}$$

depends in a deterministic way on the initial coordinates, and

$$C^{(1)}_{mn} = \int_0^{t_1} du_1 \int_0^{t_2} du_2\ \ \varepsilon_{m_2}(t_1 - u_1)\ \ \varepsilon_{n_2}(t_2 - u_2)\ \ \langle f\left(u_1\right) f\left(u_2\right) \rangle \tag{5.79}$$

With $t_<$ =min (t_1, t_2) and $K\delta(u_1 - u_2)$ we get,

$$C^{(1)}_{mn} = K \int_0^{t_<} du \;\; \varepsilon_{m_2}(t_1 - u) \;\; \varepsilon_{n_2}(t_2 - u). \tag{5.80}$$

Since we have diagonalized $\mathcal{D}$,

$$\varepsilon_{mn}(t) = \sum_k R_{mk} \; e^{-\lambda_k t} \; L_{kn}. \tag{5.81}$$

We find,

$$\mathbb{C}^1_{mn} = K \;\; \sum_{kl} R_{mk} \; L_{k_2} \; R_{nl} \; L_{l_2} \;\; \int_0^{t_<} du \;\; e^{-\lambda_k(t_1-u)} \; e^{-\lambda_l(t_2-u)} \tag{5.82}$$

At $t_1 = t_2 = t$ the integral in the last eq. will be,

$$e^{-(\lambda_k t_1 + \lambda_l t_2)} \;\; \int_0^{t_<} du \;\; e^{(\lambda_k+\lambda_l)u} = \frac{1 - e^{-(\lambda_k+\lambda_l)t}}{(\lambda_k + \lambda_l)}. \tag{5.83}$$

At $t \to \infty$ will,

$$\frac{1 - e^{-(\lambda_k+\lambda_l)t}}{(\lambda_k + \lambda_l)} = \frac{1}{(\lambda_k + \lambda_l)}. \tag{5.84}$$

5.6 Linear harmonic chain (many oscillators)

The Hamiltonian of a linear chain can be described as

$$H = \frac{1}{2} \sum_{k=0}^{d} m_k \, \dot{y}_k^2 + \frac{1}{2} K_k \, y_k^2 + \frac{1}{2} \sum_{k=0}^{d-1} Q_k \, (y_{k+1} - y_k)^2 \,, \tag{5.85}$$

where all m_k, and Q_k are positive, and all K_k non-negative[3]. This means that the inverse mass matrix is

$$\mathbf{M}^{-1} = \mathrm{diag}_0 \left(m_0^{-1}, m_1^{-1}, \ldots, m_d^{-1} \right), \tag{5.86}$$

and that the interaction matrix is

$$\begin{aligned} \mathbf{K} = {} & \mathrm{diag}_0 \left(K_0, K_1, \ldots K_{d-1}, K_d \right) + \mathrm{diag}_0 \left(Q_0, Q_1, \ldots, Q_{d-1}, 0 \right) + \\ & + \mathrm{diag}_0 \left(0, Q_0, \ldots, Q_{d-2}, Q_{d-1} \right) - \sum_{n=\pm 1} \mathrm{diag}_n \left(Q_0, Q_1, \ldots, Q_{d-1} \right). \end{aligned} \tag{5.87}$$

[3] In a model where $K_k = 0$ is allowed we should probably take $K_k = 0$ everywhere except possibly at the endpoints.

In a model with random positive parameters we may choose each of the parameters as

$$Z = \bar{Z} + \Delta Z\, \zeta, \tag{5.88}$$

where Z is any of the parameters M_k, K_k, or Q_k. Here ζ should take random values in the interval $\left[-\frac{1}{2}, \frac{1}{2}\right]$ with $\Delta Z < \bar{Z}$.

Chapter 6

Simulations in Python

Recently, gained Python programing language great interest and widespread use. It is an open source language, able for developement, has a very easy syntax to write, read and learn, since Python was written in **C** programming language.

Its an interpreted high-level programming language, which able to read quickly any changes of code. There is no need to compile the code to make it run as other languages, where compiling take long time. But Python still slow language as it need to figure out what the code does every time the code runs. Thats make it attractive for rapid application development but not for the extended numerical computations. Suitably, there is a Python scientific computing packages as **NumPy** [49] and **SciPy** [50] which provides several needed routines for this kind of numerical computations, directly from Python. Fortunately these packages are freely available for most operating systems, including Linux, OSX, and MS-Windows. In this thesis was useful to use some of these routines for solving partial differential equations discretized on a square grid to proceed for a general solution of a classical, damped oscillator with a time-dependent force.

However, the classes used to solve this problem are designed with additional topologies, geometries, and applications in mind. These classes are ***Lattice***,***LatticeFunction***, and ***LatticeOperator***. A specific application from Quantum Mechanics has been refactored to extend these classes.

6.1 Anharmonic oscillators

The anharmonic oscillator was one of the first system studied by Heisenberg when he introduced matrix mechanics, and this has been an extremely popular model to study ever since. Here we will continue to study it as an

extension of the work described in section 4.1.

We have already constructed matrices for $\hat{q}$ and $\hat{p}$, with respect to the lowest N states of the Hamiltonian

$$H_0 = \hat{p}^2 + \hat{q}^2. \tag{6.1}$$

One may construct matrices for $\hat{p}^2$ and $\hat{q}^2$ by squaring the matrices for $\hat{q}$ and $\hat{p}$, as was done in section 4.1, but it seems slightly better to construct them directly in infinite-state model, and only thereafter reduce the to the space of lowest N states. A similar procedure can be done to generate the lowest N states of $\hat{q}^4$.

Moreover, for a symmetric model,

$$H = a^{(2)}\hat{p}^2 + b^{(2)}\hat{q}^2 + b^{(4)}\hat{q}^4, \tag{6.2}$$

only matrix elements between states with the same parity will be non-zero. Hence, it is advantageous to construct separate H-matrices for the positive and negative parity subspaces. The results are given in the nex subsection

6.1.1 Explicit matrix representations

Choose units in (2.31) such that $\bar{m}\bar{\omega} = 1$ and $\hbar = 1$, and define matrices

$$Q^{(2e)}_{m,n} \equiv \left(\hat{q}^2\right)_{2m,2n}, \tag{6.3a}$$
$$Q^{(2o)}_{m,n} \equiv \left(\hat{q}^2\right)_{2m+1,2n+1}, \tag{6.3b}$$
$$P^{(2e)}_{m,n} \equiv \left(\hat{p}^2\right)_{2m,2n}, \tag{6.3c}$$
$$P^{(2o)}_{m,n} \equiv \left(\hat{p}^2\right)_{2m+1,2n+1}, \tag{6.3d}$$
$$Q^{(4e)}_{m,n} \equiv \left(\hat{q}^4\right)_{2m,2n}, \tag{6.3e}$$
$$Q^{(4o)}_{m,n} \equiv \left(\hat{q}^4\right)_{2m+1,2n+1}. \tag{6.3f}$$

The nonzero elements of the above quantities are

$$Q^{(2e)}_{n,n} = \tfrac{1}{2}(4n+1), \tag{6.4a}$$
$$Q^{(2e)}_{n,n+1} = Q^{(2e)}_{n+1,n} = \tfrac{1}{2}\sqrt{(2n+1)(2n+2)}, \tag{6.4b}$$
$$Q^{(2o)}_{n,n} = \tfrac{1}{2}(4n+3), \tag{6.4c}$$
$$Q^{(2o)}_{n,n+1} = Q^{(2o)}_{n+1,n} = \tfrac{1}{2}\sqrt{(2n+2)(2n+3)}, \tag{6.4d}$$
$$Q^{(4e)}_{n,n} = \tfrac{1}{4}(24n^2+12n+3), \tag{6.4e}$$
$$Q^{(4e)}_{n,n+1} = Q^{(4e)}_{n+1,n} = \tfrac{1}{4}(8n+6)\sqrt{(2n+1)(2n+2)}, \tag{6.4f}$$
$$Q^{(4e)}_{n,n+2} = Q^{(4e)}_{n+2,n} = \tfrac{1}{4}\sqrt{(2n+1)(2n+2)(2n+3)(2n+4)}, \tag{6.4g}$$
$$Q^{(4o)}_{n,n} = \tfrac{1}{4}(24n^2+36n+15), \tag{6.4h}$$
$$Q^{(4o)}_{n,n+1} = Q^{(4o)}_{n+1,n} = \tfrac{1}{4}(8n+10)\sqrt{(2n+2)(2n+3)}, \tag{6.4i}$$
$$Q^{(4o)}_{n,n+2} = Q^{(4o)}_{n+2,n} = \tfrac{1}{4}\sqrt{(2n+2)(2n+3)(2n+4)(2n+5)}, \tag{6.4j}$$

with further $P^{(2p)}_{nn} = Q^{(2p)}_{nn}$, $P^{(2p)}_{n,n+1} = P^{(2p)}_{n+1,n} = -Q^{((2p))}_{n,n+1}$ for parity $p = e$ or o. Note that all odd matrix elements can be obtained by shifting $n \to n + \frac{1}{2}$ in the expressions for the even ones.

6.1.2 Extensions to more than one dimension

By using the representation above, and taking $N = 50$, we found that the about lowest 20 eigenvalues of the pure anharmonic oscillator,

$$H_{\text{an}} = \hat{p}^2 + \hat{q}^4, \tag{6.5}$$

can be reduced to full numerical precision (about 14 decimals). I.e., no improvement is obtained for the lowest states by going to higher N.

This means that two-dimensional systems can be handled by taking $N = 50$ in two directions, leading to a 2500×2500-matrix. This is also unproblematic to handle by normal (dense) matrix routines, running on an ordinary laptops. By going to three dimensions one may have to use a Hilbert space of dimension $50^3 = 125\,000$. This is a bit to large for a dense matrix routine on a laptop, but can easily be handled by a sparse, iterative routine.

Even 4-dimensional systems, $50^4 = 6\,250\,000$, should be easily within reach (with some time and patience), while 5-dimensional systems, $50^5 = 312\,500\,000$, would probably require and a high-end laptop (and lot of time and patience). But these last two possibilities have not been explored.

It is fairly straightforward to write Python code which generalize matrix multiplications to multi-index object. Recall the operation of $\hat{q}$ on a one-dimensional wave-function

```
Psi[:-1] += Q11 * psi[1:]
Psi[1:]  += Q11 * psi[:-1]
```

. In a two-dimensional model, `psi` will be a two-index array. Then, the operation of $\hat{q}_x$ can simply be coded as

```
Psi[:-1,:] += Q11 * psi[1:, :]
Psi[1:, :] += Q11 * psi[:-1,:]
```

, while the corresponding operation of $\hat{q}_x$ becomes

```
Psi[:, :-1] += Q11 * psi[:, 1:]
Psi[:, 1:]  += Q11 * psi[:,:-1]
```

Then generalizations to more dimensions, and matrices with more sub- and super-diagonals in a single diminsion, is straightforward. We have use the above technique to generate `LinearOperator` representations of hamiltonians of the form

$$H = \sum_{j=1}^{D} \left(a_j^{(2)} \hat{p}_j^2 + b_j^{(2)} \hat{q}_j^2 + b_j^{(4)} \hat{q}_j^4 \right) + \sum_{1 \le j < k \le D} b_{j,k}^{(2,2)} \hat{q}_j^2 \, \hat{q}_j^2 \tag{6.6}$$

for arbitrary dimensions D, and coefficient arrays $a^{(2)}$, $b^{(2)}$, $b^{(4)}$, $b^{(2,2)}$. This code is organized as a Python class, listed in appendix C. An example using this class is listed in appendix D.

6.2 Decay of a two-state atom

Consider a decay process like the one illustrated in figure 2.2. As a simplified model for the atom we consider a two-state system with Hamiltonian

$$H_{\mathrm{a}} = \begin{pmatrix} E_a & 0 \\ 0 & 0 \end{pmatrix} = E_a \, \sigma^+ \sigma^-, \text{ where } \sigma^+ = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad \sigma^- = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}. \tag{6.7a}$$

The atom may decay to a "photon". We model the non-interacting photon system by a finite (a numerical limitation) collection of harmonic oscillators,

$$H_{\mathrm{ph}} = \sum_{n=0}^{N} E_n \, a_n^\dagger a_n. \tag{6.7b}$$

Finally we model the interaction between the atom and the photon system by an interaction Hamiltonian

$$H_{\text{int}} = \sum_{n=0}^{N} \lambda_n \left(\sigma^+ a_n + \sigma^- a_n^\dagger \right), \tag{6.7c}$$

so that the total hamiltonian becomes

$$H = H_{\text{a}} + H_{\text{ph}} + H_{\text{ph}}. \tag{6.7d}$$

This Hamiltonian is constructed such that the total "excitation number"

6.2.1 Physical model with Hamiltonian

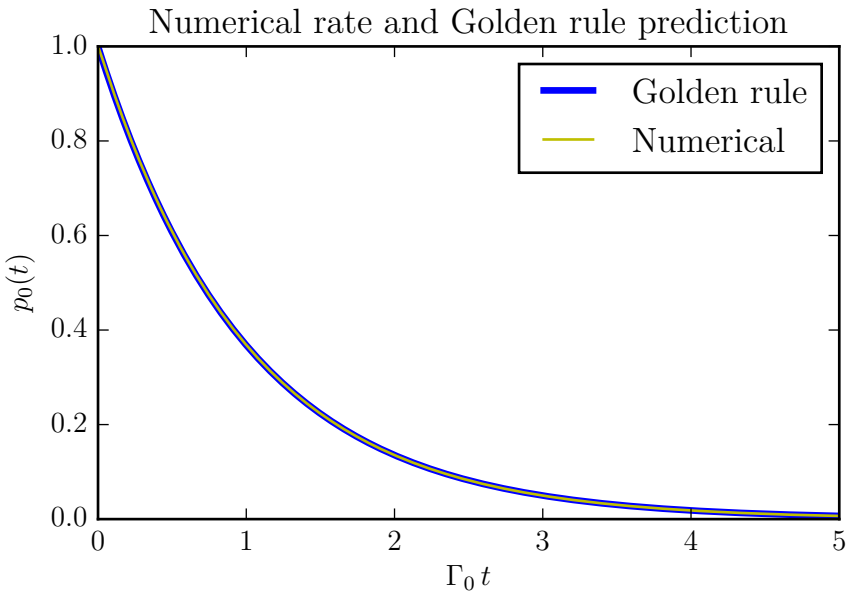


Figure 6.1: As shown in this figure, the numerical calculation of the survival probability of an excited two-level atom agrees very well with the exponential decay rate $p_0(t) = \mathrm{e}^{-\Gamma_0 t}$ predicted by the Fermi Golden rule.

$$N_e = \sigma^+ \sigma^- + \sum_{n=0}^{N} a_n^\dagger a_n \tag{6.8}$$

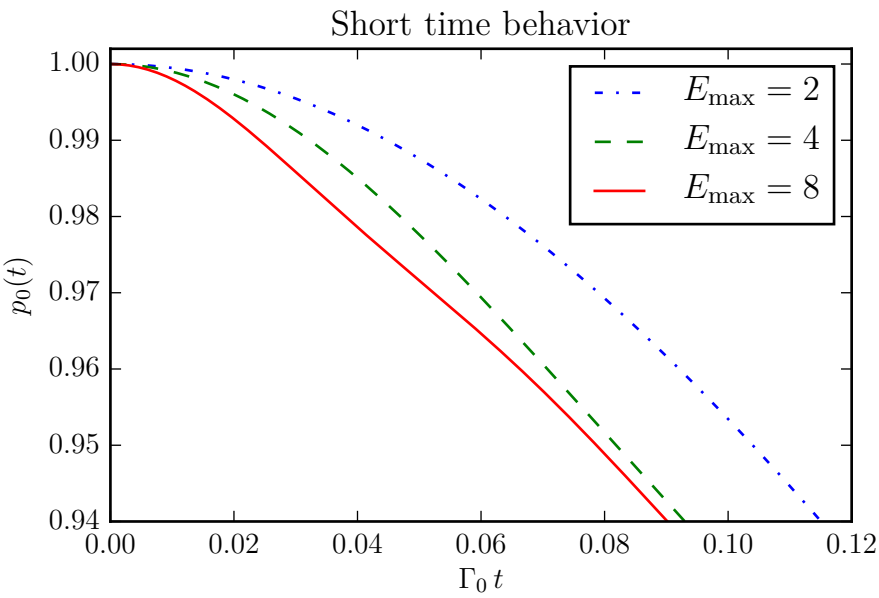


Figure 6.2: Short time behavior of the atomic decay rate. The behavior is not exponential in time t at first, but quadratic in t. This phenomenon is reffered to as the *quantum Zeno effect*. The curves are calculated for different values of $E_{\max}$, with λ adjusted such that Γ_0 remains constant. As can be seen, the quadratic range becomes smaller and smaller as $E_{\max}$ increases. For a real-world system, $E_{\max}$ is infinite; hence the effect can be expected to disappear.

is conserved, because H_{int} may reduce the atomic excitation by one (from the excited atom to the ground state) with the creation of a photon (a photonic excitation), or *vice versa*.

Since N_e commutes with H, we may reduce H to the subspace where N_e is constant. For $N_e = 1$, this leads to the $(N+2) \times (N+2)$ Hamiltonian

$$H = \begin{pmatrix} E_a & \gamma_0 & \gamma_1 & \cdots & \gamma_N \\ \gamma_0 & E_0 & 0 & \cdots & 0 \\ \gamma_1 & 0 & E_1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ \gamma_N & 0 & 0 & \cdots & E_N \end{pmatrix} \tag{6.9}$$

with all matrix elements real. We will only consider the case with n-independent $\gamma_n = \gamma/\sqrt{N}$.

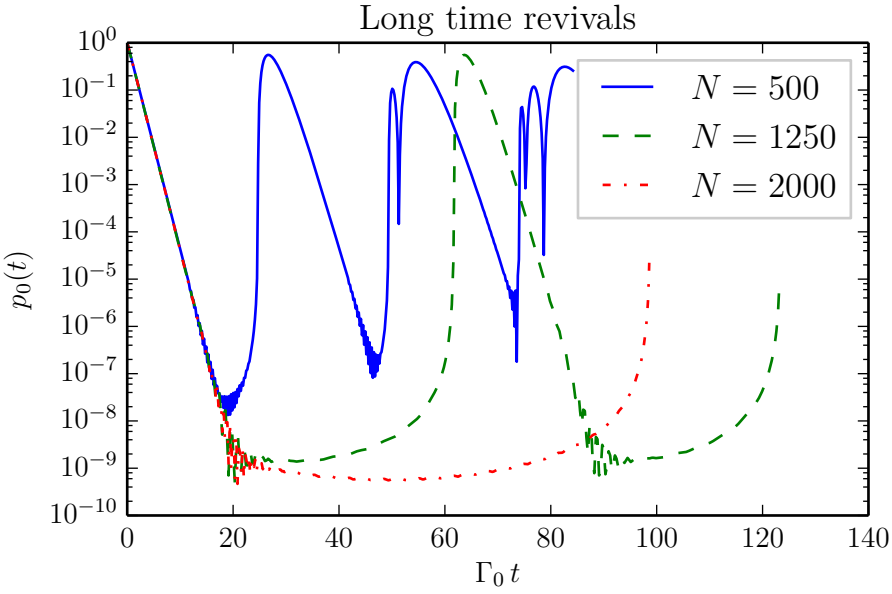


Figure 6.3: As shown in this figure, there will be a revival of the excited state afters sufficiently long time. The revival time is inversely proportional to the the gap $\Delta E = E_{n+1} - E_n$, hence proportional to N. The plateau of minimum $p_0(t) \approx 10^{-9}$ which seems to develop is probably an artifact of finite numerical accuracy.

This representation is such that the state

$$\psi^{(a)} = \big(1, 0, \ldots, 0\big)^T, \tag{6.10a}$$

represents an excited atom with energy E_a, and zero photons. The state

$$\psi^{(n)} = \big(0, \ldots, n, \ldots, 0\big)^T, \tag{6.10b}$$

represents an atom in the ground state plus exactly one photon with energy E_n. In the real world the number of photon modes is infinite, with a continuous spectrum of energies extending to infinity. But one may use this Hamiltonian (6.9) to model systems where the photons are confined to a cavity, and have a discrete spectrum.

We will use this model to solve the Schrödinger equation

$$\mathrm{i}\frac{d}{dt}\psi(t) = H\,\psi(t), \quad \psi(0) = \psi^{(a)}, \tag{6.11}$$

numerically. The procedure for doing this was outlined in section 4.2, with the complete code listed in appendix E.

6.2.2 Perturbation theory. Fermi Golden Rule

Assuming that λ is small we may calculate the decay amplitude to first order in perturbation theory, interpreting the result as a calculation of the decay rate Γ_0, defined such that the probability of finding the excited state being excited at time t is

$$p_0(t) = \mathrm{e}^{-\Gamma_0 t}, \tag{6.12}$$

provided we start in the excited state at time $t = 0$. According to the Fermi Golden rule we should in this case have

$$\Gamma_0 = 2\pi\lambda^2\rho(E_a)/N, \tag{6.13a}$$

where $\rho(E_a)$ is the density of states per energy at energy E_a. We will only consider the case that E_n is distributed evenly with n from $E_0 = 0$ to $E_N = E_{\max}$. Hence the density of states is $\rho(E_a) = N/E_{\max}$. This leads to the prediction that

$$\Gamma_0 = 2\pi\lambda^2/E_{\max}, \tag{6.13b}$$

which may be compared with the numerical results. As shown if figure 6.1 the agreement is excellent, provided one looks in a favorable time interval.

6.2.3 Quadratic behavior at short times

However, on closer look one finds deviation from exponential behavior. For short times t the decay rate is not exponential (i.e., linear in time for small t), but quadratic in t. This is easy to understand by solving the Schroödinger equation (6.11) to first order in t. We find

$$\langle\psi^{(n)}|\psi(t)\rangle = \text{-i}\lambda t/\sqrt{N},$$

such that the probability for *not* being the state $\psi^{(a)}$ becomes, cf. figure 6.2,

$$1 - p_0(t) = \lambda^2 t^2. \tag{6.14}$$

6.2.4 Long-time revival behavior

For large times we observe a different deviation from exponential decay, as shown in figure 6.3. Since the photon decays into a finite spatial region, of size inverse to the energy gap $\Delta E = E_{n+1} - E_n$, it must eventually return. The is shown by repeated revivals of the excited state after a sufficiently long times. As can be seen, the revival time increased with N, since this makes ΔE smaller. The numerical curves also exhibit a plateau of minimum survival probability, independent of N. This is probabliy an artifact of finite numerical accuracy.

6.3 Langevin description of stochastic evolution

A general mathematical formulation of such models may start from the Lagrangian

$$L = \frac{1}{2}\sum_{\boldsymbol{j}} M_{\boldsymbol{j}}\dot{\boldsymbol{q}}_{\boldsymbol{j}}^2 - \sum_{\boldsymbol{j}} U_{\boldsymbol{j}} - \frac{1}{2}\sum_{\boldsymbol{j},\boldsymbol{k}} V_{\boldsymbol{j}\boldsymbol{k}}, \tag{6.15}$$

where $U_{\boldsymbol{j}} \equiv U_{\boldsymbol{j}}(\boldsymbol{q}_{\boldsymbol{j}})$, $V_{\boldsymbol{j}\boldsymbol{k}} \equiv V_{\boldsymbol{j}\boldsymbol{k}}(\boldsymbol{q}_{\boldsymbol{j}}, \boldsymbol{q}_{\boldsymbol{k}})$, and the indices $\boldsymbol{j}$, $\boldsymbol{k}$ run over all lattice sites. The interactions $V_{\boldsymbol{j}\boldsymbol{k}} = V_{\boldsymbol{k}\boldsymbol{j}}$ are symmetric in their arguments, $V_{\boldsymbol{j}\boldsymbol{k}}(\boldsymbol{q}_{\boldsymbol{j}}, \boldsymbol{q}_{\boldsymbol{k}}) = V_{\boldsymbol{j}\boldsymbol{k}}(\boldsymbol{q}_{\boldsymbol{k}}, \boldsymbol{q}_{\boldsymbol{j}})$. The corresponding Hamiltonian

$$H = \frac{1}{2}\sum_{\boldsymbol{j}} M_{\boldsymbol{j}}^{-1}\dot{\boldsymbol{p}}_{\boldsymbol{j}}^2 + \sum_{\boldsymbol{j}} U_{\boldsymbol{j}} + \frac{1}{2}\sum_{\boldsymbol{j},\boldsymbol{k}} V_{\boldsymbol{j}\boldsymbol{k}}, \tag{6.16}$$

generate the classical equations of motion

$$\dot{\boldsymbol{q}}_{\boldsymbol{j}} = M_{\boldsymbol{j}}^{-1}\boldsymbol{p}_{\boldsymbol{j}}, \tag{6.17a}$$

$$\dot{\boldsymbol{p}}_{\boldsymbol{j}} = -\boldsymbol{\nabla}_{\boldsymbol{j}}\Big[U_{\boldsymbol{j}} + \sum_{\boldsymbol{k}} V_{\boldsymbol{j}\boldsymbol{k}}\Big]. \tag{6.17b}$$

To introduce (thermal or quantum) fluctuations into this model we convert eqs.(6.17) to a set of stochastic equations, by adding damping ($\boldsymbol{\Gamma}$) and (random) fluctuating force ($\boldsymbol{f}$) terms. This leads to the system of Langevin equations,

$$\dot{\boldsymbol{q}}_{\boldsymbol{j}} = M_{\boldsymbol{j}}^{-1}\boldsymbol{p}_{\boldsymbol{j}}, \tag{6.18a}$$

$$\dot{\boldsymbol{p}}_{\boldsymbol{j}} = -\boldsymbol{\nabla}_{\boldsymbol{j}}\Big[U_{\boldsymbol{j}} + \sum_{\boldsymbol{k}} V_{\boldsymbol{j}\boldsymbol{k}}\Big] - \boldsymbol{\Gamma}_{\boldsymbol{j}} + \boldsymbol{f}_{\boldsymbol{j}}(t), \tag{6.18b}$$

where $\boldsymbol{f}_{\boldsymbol{j}}(t)$ are gaussian distributed with zero mean, $\langle \boldsymbol{f}_{\boldsymbol{j}}(t)\rangle = 0$, and variance

$$\langle \boldsymbol{f}_{\boldsymbol{j}}(t)\boldsymbol{f}_{\boldsymbol{k}}(u)\rangle = \mathbb{G}_{\boldsymbol{j}}\,\delta_{\boldsymbol{j}\boldsymbol{k}}\,\delta(t-u). \tag{6.18c}$$

6.4 Example 1: One-dimensional harmonic oscillator

As an introductory example, consider the case of a one-dimensional harmonic oscillator, described by the Hamiltonian $H = \frac{1}{2}(p^2 + q^2)$. The Langevin equations becomes

$$\begin{pmatrix} \dot{q} \\ \dot{p} \end{pmatrix} + \begin{pmatrix} 0 & -1 \\ 1 & \gamma \end{pmatrix}\begin{pmatrix} q \\ p \end{pmatrix} = \begin{pmatrix} 0 \\ f(t) \end{pmatrix}. \tag{6.19}$$

They can be integrated to

$$\begin{pmatrix} q(t) \\ p(t) \end{pmatrix} = \mathrm{e}^{-t\mathcal{D}} \begin{pmatrix} q(t_0) \\ p(t_0) \end{pmatrix} + \int_{t_0}^{t} \mathrm{e}^{-(t-u)\mathcal{D}} \begin{pmatrix} 0 \\ f(u) \end{pmatrix} du. \tag{6.20}$$

Here $\mathcal{D} = \begin{pmatrix} 0 & -1 \\ 1 & \gamma \end{pmatrix}$, so that for real $\omega \equiv (1 - \gamma^2/4)^{1/2}$,

$$\mathrm{e}^{-t\mathcal{D}} = \mathrm{e}^{-\gamma t/2} \begin{pmatrix} \cos\omega t + \kappa \sin\omega t & \omega^{-1} \sin\omega t \\ -\omega^{-1} \sin\omega t & \cos\omega t - \kappa \sin\omega t \end{pmatrix} \tag{6.21a}$$

$$= \mathrm{e}^{-\gamma t/2} \begin{pmatrix} \rho\cos(\omega t - \phi) & \omega^{-1} \sin\omega t \\ -\omega^{-1} \sin\omega t & \rho\cos(\omega t + \phi) \end{pmatrix}, \tag{6.21b}$$

where $\kappa = \frac{1}{2}\gamma\omega^{-1} = \tan\phi$, and $\rho = (1 + \kappa^2)^{1/2} = 1/\cos\phi$.

6.5 Numerical implementation in NumPy

A numerical implementation requires very little work in **NumPy**. One first creates the dynamical matrix $\mathcal{D}$ in some way. F.i.

```
gamma = 0.1
dynamicMatrix = numpy.array([[0,-1],[1, gamma]])
```

The eigenvalues and eigenvectors are then painless to find

```
[evals, R]     = numpy.linalg.eig(dynamicMatrix)
L              = numpy.linalg.inv(R)
```

With the initial conditions specified,

```
z0 = numpy.array([0, 1])
g0 = numpy.dot(L,z0)
```

it is straightforward to evaluate the solution at a specified set of timevalues:

```
# Prepare for computation of  time evolution
tstart = 0; tend = 20; ntimes = 1001
tvals  = numpy.linspace(tstart,tend,ntimes)
qvals  = numpy.zeros(ntimes, dtype=float)
pvals  = numpy.zeros(ntimes, dtype=float)
# Evaluate the path
for n in xrange(ntimes):
t  = tvals[n]
gt = numpy.exp(-t*evals)*g0
```

```
zt = numpy.real(numpy.dot(R, gt))
qvals[n] = zt[0]
pvals[n] = zt[1]
```

Figure example of damped oscillator, plus code example

Note the code snippets above require very little change when generalized to a $2D$–dimensional linear Hamiltonian system.

Chapter 7

Summary

This thesis has reported from works done over many years, collecting notes and programs from different stages of development. For this reason it may contain some unintended repetition of content, and (equally unintended) variation of notation.

Apart from a general literature study of the formulations and development of quantum mechanics, it attempts to solve some interesting problems numerically, with the help of code available from the `NumPy` and `SciPy` packages of `Python`.

The syntax of the multi-dimensional arrays in `Numpy` is such that one often can write rather compact code for the required linear algebra operations. The most time consuming processes, like multiplication of large matrices or the solution of eigenvalue problems, are done in a compact natural expression or by a single function call. This means that such operations are carried out by precompiled library routines generated from code written in numerically efficient languages like Fortran, C or C++. The use of an interpreted computer language like Python does not prohibit efficient numerical use of the computer.

We have found that a problem like finding the lowest eigenvalues of a full 3-dimensional anharmonic oscillator (i.e., without separation of variables) can be solved on an ordinary laptop in a few minutes.

For the model of the a two-state decaying atom it was found that the limiting factor was not the available computer memory or computational time, but the inherent limitation of double precision numerical accuracy.

The numerical modelling of the classical Langevin equations did not reach a conclusive stage.

Appendix A

MatrixMechanicsDemo.py

```
#! /usr/bin/env python
# -*- coding: utf-8 -*-
# File: MatrixMechanicsDemo.py

import sys
import time
import pickle
import numpy
from scipy.linalg import eigh, eigvalsh
from scipy.sparse.linalg import LinearOperator, eigsh

def checkDiagonalization(N=5):
    t0 = time.time()
    Q11 = numpy.sqrt(numpy.arange(1, N)/2)

    def q(psi):
        Psi = numpy.zeros_like(psi)
        Psi[:-1] += Q11 * psi[1:]
        Psi[1:]  += Q11 * psi[:-1]
        return Psi

    def ip(psi):
        Psi = numpy.zeros_like(psi)
        Psi[:-1] += Q11 * psi[1:]
        Psi[1:]  -= Q11 * psi[:-1]
        return Psi
```

```
    def matrix(operator):
        M = numpy.zeros((N, )*2)        # Space for matrix
        id = numpy.eye(N)               # The unit matrix
        for n in range(N):
            M[:,n] = operator(id[:,n]) # Add n'th column
        return M

    Mq = matrix(q); Mip = matrix(ip)
    C = numpy.dot(Mq, Mip) - numpy.dot(Mip, Mq)
    # print (C)

    Mp2 = -numpy.dot(Mip, Mip); Mq2 = numpy.dot(Mq, Mq)
    H = Mp2 + Mq2
    # print (H)

    H = Mp2 + numpy.dot(Mq2, Mq2)
    # print (H)
    E, S = eigh(H)
    # print (E)
    # for n in range(N):
    #     psi = S[:, n]; dpsi = numpy.dot(H, psi) - E[n]*psi
    #     print ("%.5e" % numpy.linalg.norm(dpsi,
    ord=numpy.inf))

    D = numpy.dot(S.T, numpy.dot(H, S))
    err = numpy.linalg.norm(D - numpy.diag(E), ord=numpy.inf)
    dt = time.time() - t0
    print ("N=%4d: maxerror=%.5e  time=%8.5f secs" % (N, err,
    dt))

    # unity = numpy.dot(S.T, S)
    # ortherr = numpy.linalg.norm(unity - numpy.eye(N),
    #     ord=numpy.inf)
    # unity = numpy.dot(S, S.T)
    # comperr = numpy.linalg.norm(unity - numpy.eye(N),
    #     ord=numpy.inf)
    # print ("N=%4d: ortherror=%.5e  comperr=%.5e" %
    #     (N, ortherr, comperr))
```

```
def checkDiagonalization2(N=5):
    if N >= 3000:
        k = 100
    elif N >= 500:
        k = 80
    elif N >= 50:
        k = 20
    elif N >= 20:
        k=10
    else:
        k=6
    t0 = time.time()
    Q11 = numpy.sqrt(numpy.arange(1, N)/2)

    def q(psi):
        Psi = numpy.zeros_like(psi)
        Psi[:-1] += Q11 * psi[1:]
        Psi[1:]  += Q11 * psi[:-1]
        return Psi

    def ip(psi):
        Psi = numpy.zeros_like(psi)
        Psi[:-1] += Q11 * psi[1:]
        Psi[1:]  -= Q11 * psi[:-1]
        return Psi

    def p2(psi):
        chi = ip(psi)
        return -ip(chi)

    def q2(psi):
        chi = q(psi)
        return q(chi)

    def H(psi):
        chi = q2(psi)
        return p2(psi) + q2(chi)

    A = LinearOperator((N, N), matvec=H, dtype=float)
    E, S = eigsh(A, k=k, which='SM')
```

```
    dt = time.time() - t0
    err = -1.
    for n in range(k):
        psi = S[:, n]; dpsi = H(psi) - E[n]*psi
        errn = numpy.linalg.norm(dpsi, ord=numpy.inf)
        if errn > err:
            err = errn
    print ("N=%4d␣(k=%3d):␣maxerror=%.5e␣␣time=%9.5f␣secs" %
        (N, k, err, dt))

def main(argv):
    for N in [5, 20, 50, 200, 500, 2000, 5000]:
        checkDiagonalization(N=N)
    # for N in [50, 200, 500, 2000, 5000]:
    #     checkDiagonalization2(N=N)

if __name__ == "__main__":
    main(sys.argv[1:])
```

Appendix B

WaveMechanicsDemo.py

```
#! /usr/bin/env python
# -*- coding: utf-8 -*-
# File: MatrixMechanicsDemo.py

import sys
import time
import pickle
import numpy
from scipy.linalg import eigh

from matplotlib import rc
rc('text', usetex=True)
import matplotlib
# matplotlib.use('PDF')
import matplotlib.pyplot as pyplot

def _H(psi):
    dpsi0 = lamda * numpy.sum(psi[1:])
    psi[1:] *= en; psi[1:] += lamda * psi[0]
    psi[0]  *= e0; psi[0] += dpsi0
    return psi

def matrix(operator):
    M = numpy.zeros((N, )*2)       # Space for matrix
    id = numpy.eye(N)              # The unit matrix
    for n in range(N):
        M[:,n] = operator(id[:,n]) # Add n'th column
```

```
    return M


def main(argv):
    global N, e0, en, lamda
    N = 400; e0 = 1
    lamda = 0.12/numpy.sqrt(N-1)
    en = numpy.linspace(0, 2*e0, N-1)
    H = matrix(_H)
    E, S = eigh(H)
    tmax = 120   # 120 for vanlig plot
    Nt = 200
    t = numpy.linspace(0., tmax, Nt)
    Et = t.reshape(Nt,1) * E.reshape(1,N)      # Broadcast
   mechanism
    S02 = S[0,:]**2
    M0R = numpy.dot(numpy.cos(Et), S02)
    M0I = numpy.dot(numpy.sin(Et), S02)
    P0t = M0R**2 + M0I**2
    # pyplot.ylim(0.9, 1.01)
    pyplot.plot(t, P0t)
    pyplot.show()


if __name__ == "__main__":
    main(sys.argv[1:])
```

Appendix C

MatrixQM.py

```
#! /usr/bin/env python
# -*- coding: utf-8 -*-
# File: MatrixQM.py

import sys
import numpy
from scipy.linalg import eigh

class MatrixQM():
    '''Representation of H with respect to harmonic oscillator
    basis
    '''

    def __init__(self, N=50):
        self.N = N
        n = numpy.arange(2*self.N)
        self.Q11 = (numpy.sqrt(n+1)/2)[:-1]
        self.Q20 = (2*n+1)/2
        self.Q22 = (numpy.sqrt((n+1)*(n+2))/2)[:-2]
        self.Q40 = (6*n*n + 6*n + 3)/4
        self.Q42 = ((4*n+6)*numpy.sqrt((n+1)*(n+2))/4)[:-2]
        self.Q44 = (numpy.sqrt((n+1)*(n+2)*(n+3)*(n+4))/4)[:-4]

    def setTplusV(self, dim=1, a2=None, V=None):
        '''Set properties for Hamiltonian

            H = sum(a2[j] p_j**2; j=1..dim) + V(x)
```

```
        '''
        self.dim = dim
        self.shape = (2*self.N, ) * dim
        if a2 is None:
            self.a2 = (1,) * dim
        if (dim==1) and ((type(a2)==int) or (type(a2)==float)):
            self.a2 = (a2,)

        A = self.matrix(self.q,  shape=(2*self.N,))
        Q, S = eigh(A)
        self.Q = Q
        A = self.matrix(self.p2, shape=(2*self.N,))
        self.P2 = numpy.dot(S.T, numpy.dot(A, S))

    def setSimpleH(self, dim=1, a2=None, b2=None, b4=None,
b22=None):
        '''Set matrix elements for simple symmetric Hamiltonian

            H = sum(a2[j] p_j**2 + b2[j] q_j**2 + b4[j]
q_j**4; j=1..dim)
                + sum( b22[j,k] q_j**2 q_k**2; j=1..dim-1,
k=j+1..dim)
        '''
        self.dim = dim; self.shape = (self.N, ) * dim

        kw = [a2, b2, b4, b22]
        dfltval = [1, 1, 0, 0]
        dfltsize = [dim, dim, dim, (dim*(dim-1))//2]
        d1 = [1, 1, 1, 2]
        for k in range(len(kw)):
            if kw[k] is None:
                kw[k] = (dfltval[k], ) * dfltsize[k]
            if (dim==d1[k]) and (type(kw[k])==int or
type(kw[k])==float):
                kw[k] = (kwarg[k], )
        a2, b2, b4, self.b22 = kw

        self.H0 = [0, ]*dim; self.H2 = [0, ]*dim; self.H4 =
[0,]*dim
        self.h0 = [0, ]*dim; self.h2 = [0, ]*dim
```

```
        N = self.N; x = [1,]*dim
        for n in range(dim):
            xn = list(x); xn[n] = N; tn = tuple(xn)
            self.H0[n] = ((b2[n]+a2[n]) * self.Q20 +
                          b4[n] * self.Q40).reshape(tn)
            self.h0[n] = self.Q20.reshape(tn)

            xn[n] = N-2; tn = tuple(xn)
            self.H2[n] = ((b2[n]-a2[n]) * self.Q22 +
                          b4[n] * self.Q42).reshape(tn)
            self.h2[n] = self.Q22.reshape(tn)

            xn[n] = N-4; tn = tuple(xn)
            self.H4[n] = (b4[n] * self.Q44).reshape(tn)

    def setParities(self, parities=None):
        '''Select subspace of parity symmetries
        '''
        if parities is None:
            parities = (0, ) * self.dim
        if (self.dim==1) and type(parities)==int:
            parities = (parities, )
        self.Hs0 = [0, ]*self.dim; self.Hs1 = [0, ]*self.dim
        self.Hs2 = [0, ]*self.dim
        self.hs0 = [0, ]*self.dim; self.hs1 = [0, ]*self.dim
        for n in range(self.dim):
            if parities[n] == 0:
                self.Hs0[n] = (self.H0[n])[0::2]
                self.Hs1[n] = (self.H2[n])[0::2]
                self.Hs2[n] = (self.H4[n])[0::2]
                self.hs0[n] = (self.h0[n])[0::2]
                self.hs1[n] = (self.h2[n])[0::2]
            else:
                self.Hs0[n] = (self.H0[n])[1::2]
                self.Hs1[n] = (self.H2[n])[1::2]
                self.Hs2[n] = (self.H4[n])[1::2]
                self.hs0[n] = (self.h0[n])[1::2]
                self.hs1[n] = (self.h2[n])[1::2]

    def H(self, psi):
```

```
        '''LinearOperator representation of a simple symmetric
    Hamiltonian
        '''
        phi = psi.reshape((self.N, ) * self.dim)
        phiH = numpy.zeros_like(phi)
        t0 = [slice(None, None),] * self.dim
        for n in range(self.dim):
            phiH += self.Hs0[n] * phi
            tn = list(t0); tn[n] = slice(None, -1)
            sn = list(t0); sn[n] = slice(1, None)
            phiH[tn] += self.Hs1[n] * phi[sn]
            tn = list(t0); tn[n] = slice(1, None)
            sn = list(t0); sn[n] = slice(None,-1)
            phiH[tn] += self.Hs1[n] * phi[sn]
            tn = list(t0); tn[n] = slice(None, -2)
            sn = list(t0); sn[n] = slice(2, None)
            phiH[tn] += self.Hs2[n] * phi[sn]
            tn = list(t0); tn[n] = slice(2, None)
            sn = list(t0); sn[n] = slice(None,-2)
            phiH[tn] += self.Hs2[n] * phi[sn]

        for m in range(self.dim-1):
            chi = self.hs0[m] * phi
            tm = list(t0); tm[m] = slice(None, -1)
            sm = list(t0); sm[m] = slice(1, None)
            chi[tm] += self.hs1[m] * phi[sm]
            tm = list(t0); tm[m] = slice(1, None)
            sm = list(t0); sm[m] = slice(None, -1)
            chi[tm] += self.hs1[m] * phi[sm]
            for n in range(m+1, self.dim):
                imn = n-1 + m*(self.dim-2) - m*(m-1)//2
                phiH += (self.b22[imn] * self.hs0[n]) * chi
                hs1 = self.b22[imn] * self.hs1[n]
                tn = list(t0); tn[n] = slice(None, -1)
                sn = list(t0); sn[n] = slice(1, None)
                phiH[tn] += hs1 * chi[sn]
                tn = list(t0); tn[n] = slice(1, None)
                sn = list(t0); sn[n] = slice(None,-1)
                phiH[tn] += hs1 * chi[sn]
```

```
        return numpy.ravel(phiH)

    def q(self, psi):
        '''LinearOperator representation of position operator q
        '''
        psiQ = numpy.zeros_like(psi)
        psiQ[:-1] += self.Q11 * psi[1:]
        psiQ[1:]  += self.Q11 * psi[:-1]
        return psiQ

    def ip(self, psi):
        '''LinearOperator representation of momentum operator
    sqrt(-1)*p
        '''
        psiP = numpy.zeros_like(psi)
        psiP[:-1] += self.Q11 * psi[1:]
        psiP[1:]  -= self.Q11 * psi[:-1]
        return psiP

    def q2(self, psi):
        '''LinearOperator representation of V = q**2
        '''
        psiV = numpy.zeros_like(psi)
        psiV = self.Q20 * psi
        psiV[:-2] += self.Q22 * psi[2:]
        psiV[2:]  += self.Q22 * psi[:-2]
        return psiV

    def p2(self, psi):
        '''LinearOperator representation of T = p**2
        '''
        psiT = numpy.zeros_like(psi)
        psiT = self.Q20 * psi
        psiT[:-2] -= self.Q22 * psi[2:]
        psiT[2:]  -= self.Q22 * psi[:-2]
        return psiT

    def TplusV(self, psi):
        '''LinearOperator representation of p**2 + V(q)
        '''
```

```
        phi = psi.reshape(self.shape)
        phiH = self.V * phi
        if self.dim == 1:
            phiH += self.a2[0] * numpy.dot(self.P2, phi)
        elif self.dim == 2:
            phiH += self.a2[0] * numpy.dot(self.P2, phi)
            phiH += self.a2[1] * numpy.dot(self.P2, phi.T).T
        else:
            for n in range(self.dim):
                chi0  = numpy.dot(self.P2, numpy.swapaxes(phi,
n, -2))
                chi1  = numpy.swapaxes(chi0, 0, -2)
                phiH += self.a2[n] * numpy.swapaxes(chi1, n,
-2)
        return numpy.ravel(phiH)

    def matrix(self, operator, shape=None):
        '''Return matrix representation of operator
        '''
        if shape is None:
            shape = self.shape
        N = numpy.prod(shape)
        matrix = numpy.zeros((N, )*2) # Allocate space for
matrix
        id = numpy.eye(N)
        for n in range(N):
            psi = id[:,n].reshape(shape)
            matrix[:,n] = operator(psi)
        return matrix
```

Appendix D

useMatrixQM.py

```
#! /usr/bin/env python
# -*- coding: utf-8 -*-
# File: useMatrixQM.py

import sys
import time
import pickle
import numpy
from scipy.linalg import eigvalsh
from scipy.sparse.linalg import LinearOperator, eigsh

from MatrixQM import MatrixQM

from matplotlib import rc
rc('text', usetex=True)
import matplotlib
# matplotlib.use('PDF')
import matplotlib.pyplot as pyplot


# Eigenvalues E_n, n=0,..1000, for anharmonic oscillator
eigvals = numpy.loadtxt("x4En.dat")

def checkClass():
    marks=['^g', '^b','vr','vg','vb','or','og','ob', 'og',
    'or']
    Nvals = range(20, 56, 5)
```

```
    lenN = len(Nvals)
    evalErrors = numpy.zeros((6, lenN))
    for k in range(lenN):
        myQM = MatrixQM(N=Nvals[k])
        H = myQM.matrix(myQM.lin1De)
        errors = numpy.abs(eigvalsh(H)[:6] -
    eigvals[0:12:2][:,1])
        evalErrors[:,k] = errors
    xvals = 2*numpy.arange(6) + 1
    for k in range(lenN):
        pyplot.semilogy(xvals, evalErrors[:,k], marks[k],
            label=r"N=%d" % Nvals[k])
    pyplot.semilogy(xvals, evalErrors[:,-1], '-b')
    pyplot.xlim(-0.5, 11.5)
    pyplot.legend(loc='upper left', numpoints=3)
    pyplot.show()
#        print ("%2d: %19.16f" % (2*n, evals[n]- evalsE[n]))

def checkEvals(nmax=50):
    for N in [8, 16, 32, 64]:
        myQM = MatrixQM(N=N)
        evals = numpy.zeros(2*N)
        H = myQM.matrix(myQM.H1De)
        evals[0:2*N:2] = eigvalsh(H)
        H = myQM.matrix(myQM.H1Do)
        evals[1:2*N:2] = eigvalsh(H)
        nmx = min(2*N, nmax+1)
        pyplot.semilogy(eigvals[:nmx,0], evals[:nmx], 'g^')
    pyplot.semilogy(eigvals[:nmax+1,0],
    eigvals[:nmax+1,1],'ro')
    pyplot.xlim(-0.5, nmax+0.5)
    pyplot.ylim(0.5, 1e3)
    pyplot.show()

def checkEvalErrors(nmax=80):
    for N in [8, 16, 32, 64, 128, 256]:
        myQM = MatrixQM(N=N)
        evals = numpy.zeros(2*N)
        H = myQM.matrix(myQM.H1De)
        evals[0:2*N:2] = eigvalsh(H)
```

```
        H = myQM.matrix(myQM.H1Do)
        evals[1:2*N:2] = eigvalsh(H)
        nmx = min(2*N, nmax+1)
        pyplot.semilogy(eigvals[:nmx,0],
    numpy.abs(evals[:nmx]-eigvals[:nmx,1]), 'g^')
    pyplot.xlim(-0.5, nmax+0.5)
#    pyplot.ylim(0.5, 1e3)
    pyplot.show()

def checkQ2(N=100):
    myQM = MatrixQM(N=N, a2=0, b2=1, b4=0)
    A = myQM.matrix(myQM.q, N=2*N)
    q1 = eigvalsh(A)
    A = myQM.matrix(myQM.H1De)
    q2e = numpy.sqrt(eigvalsh(A))
    A = myQM.matrix(myQM.H1Do)
    q2o = numpy.sqrt(eigvalsh(A))
    for n in range(N):
        print ("%22.15e" % (q1[N+n]-q2e[n]))
    #pyplot.semilogy(x, 'b.')
    #pyplot.show()

def checkQ(N=100):
    myQM = MatrixQM(N=N)
    A = myQM.matrix(myQM.q2e)
    q2, S = eigh(A)
    A = myQM.matrix(myQM.p2e)
    p2 = numpy.dot(S.T,numpy.dot(A, S))
    print (p2)

def makeV(q2):
    V = numpy.empty_like(q2)
    for n in range(q2.size):
        V[n] = q2[n]**2
    return V

def test(N=64):
    myQM = MatrixQM(N=64)
    myQM.setSimpleH1D(b2=0, b4=1)
    myQM.V = makeV(myQM.q2e)
```

```
    myQM.p2 = myQM.p2e
    A = myQM.matrix(myQM.H1De)
    evals0 = eigvalsh(A)
    A = myQM.matrix(myQM.linOpH)
    evals1 = eigvalsh(A)
    print ((evals0-evals1)[:50])

def checkHarmonicOscillators():
    '''Compute spectra of simple harmonic oscillators in 1, 2,
    3 dimensions
    '''
    myQM = MatrixQM(N=8)
    for dim in range(1, 4):
        myQM.setSimpleH(dim=dim)
        for parities in numpy.ndindex((2,) * dim):
            myQM.setParities(parities)
            t0 = time.time()
            A = myQM.matrix(myQM.H)
            eN = eigvalsh(A)
            dt = time.time() - t0
            print ("%d-dim, parity %s, eigenvalues in %f
    secs:" %
                (dim, parities, dt))
            print (eN)
        print ()

def checkAnharmonicOscillators():
    '''Compute spectra of simple anharmonic oscillators in 1,
    2, 3 dimensions
    '''
    myQM = MatrixQM(N=50)
    eNs = []
    for dim in range(1, 3):
        myQM.setSimpleH(dim=dim, b2=(0, )*dim, b4=(1, )*dim)
        for parities in numpy.ndindex((2,) * dim):
            myQM.setParities(parities)
            t0 = time.time()
            Nd = myQM.N**dim
            if Nd <= 4000:
                A = myQM.matrix(myQM.H)
```

```
                eN = eigvalsh(A)
            else:
                A = LinearOperator((Nd, Nd), matvec=myQM.H,
    dtype=float)
                eN = eigsh(A, k=20, which='SA',
    return_eigenvectors=False)
                eN = numpy.sort(eN)
            dt = time.time() - t0
            print ("%d-dim, parity %s, eigenvalues in %f
    secs:" %
                (dim, parities, dt))
            print (eN[:20])
        eNs.append([dim, parities, dt, eN])
    with open("EvalsAnharmonicOscillator.pkl", "wb") as
    outfile:
        pickle.dump(eNs, outfile)


def main(argv):
    checkTensor()
    return
    shape = (3,5,7)
    a = numpy.arange(numpy.prod(shape)).reshape(shape)
    lens = len(shape)
    if (lens > 1):
        for n in range(len(shape)):
            N = shape[n]
            p2 = numpy.arange(N**2).reshape(N,N)
            y = numpy.dot(p2, numpy.swapaxes(a,-2,n))
            print (y.shape)
            z = numpy.swapaxes(y,0,-2)
            print (z.shape)
            w = numpy.swapaxes(z,-2,n)
            print (w.shape)
            print ()
    return
    # checkHarmonicOscillators()
    # checkAnharmonicOscillators()

if __name__ == "__main__":
```

```
    main(sys.argv[1:])
```

Appendix E

TwoLevelAtom.py

```
#! /usr/bin/env python
# -*- coding: utf-8 -*-
# File: TwoLevelAtom.py

# Model for a two-level atom coupled to a "photon" distribution

import sys
import numpy
from scipy.linalg import eigh

from matplotlib import rc
rc('text', usetex=True)
import matplotlib
matplotlib.use('PDF')
import matplotlib.pyplot as pyplot

class TwoLevelAtom():
    '''A two-level atom coupled to a "photon" distribution
    '''
    def __init__(self, N=1000, lamda=0.1, E0=1, Emax=None,
    noise=0):
        '''Set model parameters
        '''
        self.N = N
        self.shape = (N+2,)
        self.E0 = E0
        if Emax is None:
```

```
            self.Emax =  8*E0
        else:
            self.Emax = Emax
        # Photon spectrum
        self.Eph = numpy.linspace(0, self.Emax, N+1)
        self.dEph = self.Emax/N # Inverse density of states
        self.lamda = lamda*numpy.sqrt(self.dEph)
        # Random perturbation of spectrum (0 <= noise < 1)
        if noise != 0:
            self.Eph +=
    noise*(numpy.random.rand(N+1)-1/2)*self.dEph

    def H(self, psi):
        '''LinearOperator representation of Hamiltonian
        '''
        dpsi0 = self.lamda * numpy.sum(psi[1:])
        psi[1:] *= self.Eph
        psi[1:] += self.lamda * psi[0]
        psi[0] *= self.E0
        psi[0] += dpsi0
        return psi

    def matrix(self, operator, shape=None):
        '''Return matrix representation of operator
        '''
        if shape is None:
            shape = self.shape
        N = numpy.prod(shape)
        matrix = numpy.zeros((N, )*2) # Allocate space for
    matrix
        id = numpy.eye(N)
        for n in range(N):
            psi = id[:,n].reshape(shape)
            matrix[:,n] = operator(psi)
        return matrix

def demonstrateModel(tmax=700, Nt=200):
    '''Demonstrate basic (favorable) properties of the model
    '''
    tL = TwoLevelAtom()
```

```
    H = tL.matrix(tL.H)
    E, S = eigh(H)
    N = len(E)
    t = numpy.linspace(0., tmax, Nt)
    Et = t.reshape(Nt,1) * E.reshape(1,N)  # Broadcast
    mechanism
    S02 = S[0,:]**2
    M0R = numpy.dot(numpy.cos(Et), S02)
    M0I = numpy.dot(numpy.sin(Et), S02)
    P0t = M0R**2 + M0I**2
    G0 = 2*numpy.pi*tL.lamda**2/tL.dEph    # Golden Rule decay
    rate
    # Plot result
    fig = pyplot.figure()
    linewidth = 358.50475/72
    fig.set_size_inches(linewidth, 0.618*linewidth)
    subfig = fig.add_subplot(1,1,1)
    pyplot.plot(G0*t, numpy.exp(-G0*t), 'b', lw=2.5,
        label=r"Golden␣rule")
    pyplot.plot(G0*t, P0t, 'y', lw=1,
        label=r"Numerical")
    pyplot.xlim(0, 5)
    pyplot.legend(loc='upper␣right')
    pyplot.title(r"Numerical␣rate␣and␣Golden␣rule␣prediction")
    pyplot.xlabel(r"$\Gamma_0\,t$")
    pyplot.ylabel(r"$p_0(t)$")
    pyplot.savefig("TwoLevelAtomicDecay0",
        dpi=1200, bbox_inches='tight')

def shortTimeBehavior(tmax=1, Nt=200):
    '''Show non-exponential behavior at short times
    '''
    Emx = [2, 4, 8];
    tmx = [1.9, 1.7, 1.6]
    mrk = ['b-.', 'g--', 'r']
    fig = pyplot.figure()
    linewidth = 358.50475/72
    fig.set_size_inches(linewidth, 0.618*linewidth)
    subfig = fig.add_subplot(1,1,1)
    for n in range(len(Emx)):
```

```
        Emax = Emx[n]; tmax = tmx[n]
        tL = TwoLevelAtom(Emax=Emax)
        G0 = 2*numpy.pi*tL.lamda**2/tL.dEph
        H = tL.matrix(tL.H)
        E, S = eigh(H)
        N = len(E)
        t = numpy.linspace(0., tmax, Nt)
        Et = t.reshape(Nt,1) * E.reshape(1,N)
        S02 = S[0,:]**2
        M0R = numpy.dot(numpy.cos(Et), S02)
        M0I = numpy.dot(numpy.sin(Et), S02)
        P0t = M0R**2 + M0I**2
        pyplot.plot(G0*t, P0t, mrk[n],
            label=r"$E_{\text{max}}␣=␣%d$" % Emax)
    pyplot.legend(loc='upper␣right')
    pyplot.title(r"Short␣time␣behavior")
    pyplot.xlabel(r"$\Gamma_0\,t$")
    pyplot.ylabel(r"$p_0(t)$")
    pyplot.ylim(0.94, 1.002)
#    pyplot.show()
    pyplot.savefig("TwoLevelAtomicDecay1",
        dpi=1200, bbox_inches='tight')

def longTimeRevival(tmax=1500, Nt=413):
    '''Show revival behavior at long times
    '''
    fig = pyplot.figure()
    linewidth = 358.50475/72
    fig.set_size_inches(linewidth, 0.618*linewidth)
    subfig = fig.add_subplot(1,1,1)
    tx = [1340, 1961, 1570]
    Nn = [500, 1250, 2000]
    mrk = ['b', 'g--', 'r-.']
    for n in range(len(Nn)):
        N = Nn[n]
        tmax = tx[n]
        tL = TwoLevelAtom(N=N)
        H = tL.matrix(tL.H)
        E, S = eigh(H)
        NE = len(E)
```

```
        t = numpy.linspace(0., tmax, Nt)
        Et = t.reshape(Nt,1) * E.reshape(1,NE)
        S02 = S[0,:]**2
        M0R = numpy.dot(numpy.cos(Et), S02)
        M0I = numpy.dot(numpy.sin(Et), S02)
        P0t = M0R**2 + M0I**2
        G0 = 2*numpy.pi*tL.lamda**2/tL.dEph
        # Plot result
        pyplot.semilogy(G0*t, P0t, mrk[n],
            label=r"$N=%d$" % N)
    leg = pyplot.legend(loc='upper right')
    leg.get_frame().set_alpha(0.2)
    pyplot.title(r"Long time revivals")
    pyplot.xlabel(r"$\Gamma_0\,t$")
    pyplot.ylabel(r"$p_0(t)$")
    pyplot.xlim(0, 140)
#    pyplot.show()
    pyplot.savefig("TwoLevelAtomicDecay2",
        dpi=1200, bbox_inches='tight')

def main(argv):
     demonstrateModel()
 #   shortTimeBehavior()
 #   longTimeRevival()
    return

if __name__ == "__main__":
    main(sys.argv[1:])
```

Appendix F

DynamicalLattice.py

```
#! /usr/bin/env python
# -*- coding: utf-8 -*-
# File: dynamicallattice.py

import sys
import time
import itertools
import numpy
from numpy import roll
import numpy.core.umath_tests as ut
import scipy
from scipy.linalg import eigvals
from scipy.sparse.linalg import LinearOperator, eigs

from lattice import Lattice
from latticefunctions import LatticeFunctions
from latticeoperator import LatticeOperator

class DynamicalLattice(LatticeOperator):
    """Dynamical evolution of linear lattice vibrations"""

    def __init__(self, lattice, Mx=None, M=None, Gx=None,
    G=None, def_K=None):

        if def_K is None:
            def_K = lambda k2: sum(k2) # Discrete Laplace
    operator in k-space
```

```
        LatticeOperator.__init__(self, lattice, stensil=None)
        self.size = 2 * lattice.size

        if Mx is None:
            # Possible values of inverse mass
            Mx = numpy.array([numpy.sqrt(1/2),1,numpy.sqrt(2)])
        if M is None:
            # Choose values drawn at random from Mx
            self.M = Mx[numpy.random.randint(Mx.size,
                size=lattice.size)].reshape(lattice.shape)
        if Gx is None:
            # Possible values of dissipatrion
            Gx = numpy.array([0.1])
        if G is None:
            # Choose values drawn at random from Gx
            self.G = Gx[numpy.random.randint(Gx.size,
                size=lattice.size)].reshape(lattice.shape)
        if def_K is None:
            def_K = lambda k2: sum(k2) # Discrete Laplace
    operator in k-space
        self.myF = LatticeFunction(lattice, def_G=def_K,
    evalG=True)

#         if dynM is None:
#             dynM = numpy.random.rand(2,2)
#             dynM[0,0] = 0
#             dynM[1,0] = -dynM[1,0]
#             dynM[1,1] = -0.1*dynM[1,1]
#             self.dynM = dynM
#         else:
#             myF = LatticeFunction(lattice)
#             # Generate masses (diagonal terms)
#             self.invM = myF.evalFr(def_F=dynM[0],
    returnResult=True)
#             # Generate dampings (local terms)
#             self.gamma = myF.evalFr(def_F=dynM[1],
    returnResult=True)
#             # Generate local interactions ()
#             self.locK = myF.evalFr(def_F=dynM[2],
```

```
    returnResult=True)
#             dynM = numpy.zero(2,2)
#             dynM[0,1] = -self.invM
#             dynM[1,1] = -self.gamma
#             dynM[1,0] = self.locK
#             self.dynM = dynM
#         self.shapeI = shapeI
#         if shapeI is None:
#             self.size = self.lattice.size
#         else:
#             self.size = self.lattice.size *
    numpy.prod(shapeI)
#         self.varOp = self.dynOp

    def linOp(self, phi0):
        """Convert input to multidimensional and output back"""
        shape = (2, ) + self.lattice.shape
        print (self.size)
        print (phi0.shape)
        return
        phi = phi0.reshape(shape)
        return numpy.ravel(self.varOp(phi))

    def varOp(self, phi):
        """Apply dynamical matrix to input vector"""
        Phi = numpy.zeros_like(phi)
        Phi[0] = self.M * phi[0]
        print ("Phi_0", Phi[0])
        Phi[1] = self.myF.FFT(phi[0])
        print ("FFT Phi_1", Phi[1])
        Phi[1] *= self.myF.valuesG
        print ("FFT Phi1*G", Phi[1])
        Phi[1] = self.myF.iFFT(Phi[1])
        print ("iFFT Phi1", Phi[1])
        Phi[1] += self.G * phi[1]
        print (Phi[1])
        return (Phi)

def checkDynamicalSystem():
```

```
    myL = Lattice(shape=(8,))
    myD = DynamicalLattice(myL)
    matD = myD.matrix()
    return
    print (matD)
    return
    matrix = myD.matrix()
    evals = eigvals(matrix)
    print (evals)

def tstGeneralizedUfunc():
    shape = (10,10)
    shapeD = shape + (2,2)
    shapeV = shape + (2,1)
    D = numpy.random.rand(*shapeD)
    V = numpy.random.rand(*shapeV)
    W = ut.matrix_multiply(D, V) # Matrix-vector mult on each
   lattice point
    for ix in numpy.ndindex(shape):
        Wix = numpy.dot(D[ix],V[ix]).reshape((2,1))
        print (numpy.max(W[ix]-Wix))

def rectFunc():
    n = 3; k=3
    myL = Lattice(shape=(2**n,)*2)
    lat = 0.5*numpy.ones(myL.shape)
    lat[2**(n-1)-k:2**(n-1)+k, 2**(n-1)-k:2**(n-1)+k] += 1.5
    print (lat)

def main(argv):
    rectFunc()
#    tstGeneralizedUfunc()
   #  checkDynamicalSystem()

if __name__ == "__main__":
    main(sys.argv[1:])
```

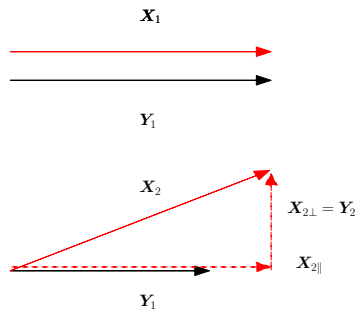
Appendix G

The Gram-Schmidt Process

The orthonormal basis is the best basis to decompose vectors. And by Gram-Schmidit process we are able to convert an arbitary basis $\{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n\}$ for any independent space V to an orthogonal basis $\{\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n\}$, then convert the orthogonal basis to an orthonormal basis $\{\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_n\}$.

The whole idea is recursive as, $\boldsymbol{Y}_j$ equal to part of the corresponding $\boldsymbol{X}_j$, which is perpendicular to all the privious $\boldsymbol{Y}_n = \{\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{j-1}\}$ vectors.

For more clarification see the illustrations below,

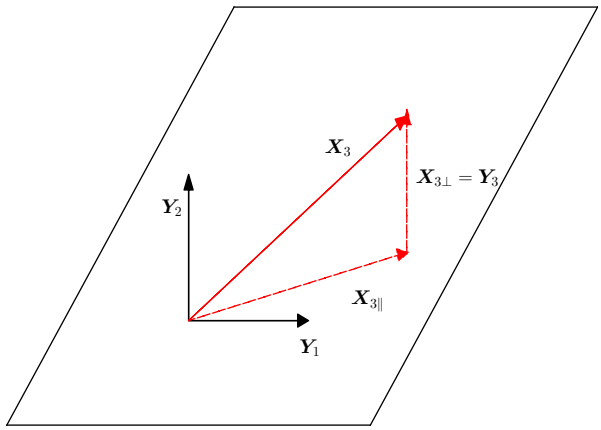


Starting from $\boldsymbol{X}_1$ and subtract it with the the previous direction which is in this stage is $\boldsymbol{Y}_1$.

$$\boldsymbol{Y}_1 = \boldsymbol{X}_1$$

Next subtract $\boldsymbol{X}_{2\parallel}$ from $\boldsymbol{Y}_1$ to get,

$$\boldsymbol{Y}_2 = \boldsymbol{X}_2 - P_{\boldsymbol{Y}_1}\boldsymbol{X}_2 = |\boldsymbol{X}_2\rangle - \frac{|\boldsymbol{Y}_1\rangle \langle \boldsymbol{Y}_1|\boldsymbol{X}_2\rangle}{\langle \boldsymbol{Y}_1|\boldsymbol{Y}_1\rangle}$$



Then subtract the $\boldsymbol{X}_{3\|}$ from $\boldsymbol{Y}_1, \boldsymbol{Y}_2$ subspace to get,

$$\boldsymbol{Y}_3 = \boldsymbol{X}_3 - P_{\boldsymbol{Y}_1}\boldsymbol{X}_3 - P_{\boldsymbol{Y}_2}\boldsymbol{X}_3$$

In next stage we subtract $\boldsymbol{X}_{4\|}$ from the subspace $(\boldsymbol{Y}_1, \boldsymbol{Y}_2, \boldsymbol{Y}_3)$ to get $\boldsymbol{Y}_4$ and so on. The vectors $\boldsymbol{Y}_1, \boldsymbol{Y}_2, \boldsymbol{Y}_3$ have to be orthogonal to each other.

Bibliography

[1] Albert Einstein, "Zur Elektrodynamik bewegter Körper. [On the Electrodynamics of Moving Bodies]," *Annalen der Physik*, vol. 322, no. 10, pp. 891–921, 1905.

[2] Richard, P, "Feynman, The character of physical law," 1965.

[3] Richard, P, "Feynman. The Feynman Lectures on Physics," *Vol. I-III*, vol. 1965, 1963.

[4] Feynman, Richard P and Leighton, Robert B and Sands, Matthew, "Lectures on Physics, vol. III," 1965.

[5] John Von Neumann, *Mathematical foundations of quantum mechanics*. No. 2, Princeton university press, 1955.

[6] Emch, GG, "Algebraic methods in statistical mechanics and quantum field theoryWiley," *New York*, 1972.

[7] Reed, M, "B. Simon Methods of Mathematical Physics IV," 1978.

[8] Teschl, Gerald, "Mathematical Methods in Quantum Mechanics," *With Applications to*, 1999.

[9] Wikipedia, The Free Encyclopedia, "Principle of least action." `https://en.wikipedia.org/wiki/Principle_of_least_action` Online; accessed 7-May-2016.

[10] Wikipedia, The Free Encyclopedia, "Hamiltonian mechanics." `https://en.wikipedia.org/wiki/Hamiltonian_mechanics` Online; accessed 1-May-2016.

[11] Wikipedia, The Free Encyclopedia, "Bohr model." `https://en.wikipedia.org/wiki/Bohr_model` Online; accessed 28-April-2016.

[12] Niels Bohr, "On the Constitution of Atoms and Molecules, Part I. Binding of Electrons by Positive Nuclei," *Philosophical Magazine*, vol. 26, no. 151, pp. 1–24, 1913.

[13] Niels Bohr, "On the Constitution of Atoms and Molecules, part II. Systems Containing Only a Single Nucleus," *Philosophical Magazine*, vol. 26, no. 153, pp. 476–502, 1913.

[14] Niels Bohr, "On the Constitution of Atoms and Molecules, part III: Systems Containing Several Nuclei," *Philosophical Magazine*, vol. 26, no. 155, pp. 857–875, 1913.

[15] Hans Geiger and Ernest Marsden, "On a Diffuse Reflection of the α-Particles," *Proceedings of the Royal Society of London A*, vol. 82, no. 557, pp. 495–500, 1909.

[16] Hans Geiger and Ernest Marsden, "The Laws of Deflexion of α Particles through Large Angles," *Philosophical Magazine*, vol. 25, no. 148, pp. 604–623, 1913.

[17] Ernest Rutherford, "The Scattering of α and β Particles by Matter and the Structure of the Atom," *Philosophical Magazine*, vol. 21, no. 155, pp. 669–688, 1911.

[18] Albert Einstein, "Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristichen Gesichtspunkt. [On a Heuristic Point of View about the Creation and Conversion of Light]," *Annalen der Physik*, vol. 322, no. 6, pp. 132–148, 1905.

[19] Johann Jacob Balmer, "Notiz über die Spectrallinien des Wasserstoffs," *Annalen der Physik*, vol. 261, no. 5, pp. 80–87, 1885.

[20] Friedrich Pashen, "Zur Kenntnis ultraroter Linenspektra. I. (Normalwellenlängen bis 2700 Å.-E.)," *Annalen der Physik*, vol. 332, no. 13, pp. 537–570, 1908.

[21] Sommerfeld, Arnold and Brose, Henry Herman Leopold Adolf, *Atomic structure and spectral lines*. Methuen, 1923.

[22] William Wilson, "The quantum-theory of radiation and line spectra," *Philosophical Magazine*, vol. 29, no. 174, pp. 795–802, 1915.

[23] Arnold Sommerfeld, "Zur Quantentheorie der Spectrallinien," *Annalen der Physik*, vol. 356, no. 17, pp. 1–94, 1916.

[24] Werner Heisenberg, "Über quantentheoretische Umdeutung kinematischer und mechanischer Beziehungen [Quantum-theoretical reinterpretation of kinematic and mechanical relations]," *Zeitschrift für Physik*, vol. 33, no. 1, pp. 879–893, 1925.

[25] Max Born and Pascual Jordan, "Zur Quantenmechanik," *Zeitschrift für Physik*, vol. 34, no. 1, pp. 858–888, 1925.

[26] Paul A.M. Dirac, "The Fundamental Equations of Quantum Mechanics," *Proc. Roy. Soc. A*, vol. 109, no. 752, pp. 642–653, 1925.

[27] Max Born, Werner Heisenberg, and Pascual Jordan, "Zur Quantenmechanik II," *Zeitschrift für Physik*, vol. 35, no. 8, pp. 557–615, 1926.

[28] Max Born, "Zur Quantenmechanik der Stoßvorgänge [On the quantum mechanics of collisions]," *Zeitschrift für Physik*, vol. 37, no. 12, pp. 863–367, 1926.

[29] Wolfgang Pauli, "Über die Wasserstoffspektrum vom Standpunkt der neuen Quantenmechanik," *Zeitschrift für Physik*, vol. 36, no. 5, pp. 336–363, 1926.

[30] Erwin Schrödinger, "An Undulatory Theory of the Mechanics of Atoms and Molecules," *Physical Review*, vol. 28, no. 6, pp. 1049–1070, 1926.

[31] Louis de Broglie, "Recherches sur la Théorie des Quanta [Researches on the Quantum Theory]," *Annales de Physique (Paris)*, vol. 3, no. 10, pp. 22–128, 1925.

[32] Wikipedia, The Free Encyclopedia, "Interaction picture."
`https://en.wikipedia.org/wiki/Interaction_picture`
Online; accessed 25-December-2015.

[33] Albert Messiah, "Quantum Mechanics (Vol. I), English translation from French by GM Temmer, fourth printing 1966."

[34] Sakurai, Jun John, "San Fu Tuan. Modern quantum mechanics," 1994.

[35] Wikipedia, The Free Encyclopedia, "Pilot wave."
`https://en.wikipedia.org/wiki/Pilot_wave`
Online; accessed 3-May-2016.

[36] Wikipedia, The Free Encyclopedia, "De Broglie-Bohm theory: Revision history."
`https://en.wikipedia.org/wiki/De_Broglie Bohm_theory`
Online; accessed 8-May-2016.

[37] Bohm, David, "A suggested interpretation of the quantum theory in terms of" hidden" variables. I," *Physical Review*, vol. 85, no. 2, p. 166, 1952.

[38] Towler, M, "De Broglie-Bohm Pilot-Wave Theory and the Foundations of Quantum Mechanics, graduate lecture course," *University of Cambridge*, 2009.

[39] Wigner, Eugene, "On the quantum correction for thermodynamic equilibrium," *Physical Review*, vol. 40, no. 5, p. 749, 1932.

[40] Wikipedia, The Free Encyclopedia, "Path integral formulation: Revision history."
`https://en.wikipedia.org/wiki/Path_integral_formulation`
Online; accessed 7-May-2016.

[41] Feynman, RP and Hibbs, AR, "Quantum mechanics and path integrals," 1965.

[42] Feynman, Richard Phillips, "Space-time approach to non-relativistic quantum mechanics," *Reviews of Modern Physics*, vol. 20, no. 2, p. 367, 1948.

[43] Wikipedia, The Free Encyclopedia, "Spectral theorem."
`https://en.wikipedia.org/wiki/Spectral_theorem`
Online; accessed 30-April-2016.

[44] "Spectral Representations." http://www.stat.tamu.edu/~suhasini/teaching673/spectral_representations.pdf.

[45] Dieudonné, Jean, *History of functional analysis*. Elsevier, 1983.

[46] C. E. Moore, "DIAGONALIZATION OF A HERMITIAN MATRIX." https://www.morehouse.edu/facstaff/cmoore/Diagonalization.htm.

[47] Parisi, G. and Wu, Yong-shi, "Perturbation Theory Without Gauge Fixing," *Sci. Sin.*, vol. 24, p. 483, 1981.

[48] Wikipedia, The Free Encyclopedia, "Acoustic wave equation."
`https://en.wikipedia.org/wiki/Acoustic_wave_equation`.

[49] Van Der Walt, Stefan and Colbert, S Chris and Varoquaux, Gael, "The NumPy array: a structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.

[50] Oliphant, Travis E, "Python for scientific computing," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 10–20, 2007.