Palacký University Olomouc
Faculty of Science

Department of Optics

Elementary simulator of single qubit dynamics

Bachelor's Thesis

Marcel Drdla

Palacký University Olomouc
Faculty of Science

Department of Optics

Elementary simulator of single qubit dynamics

Bachelor's Thesis

Author:	Marcel Drdla
Study programme:	B1701 Physics
Field of study:	Applied Physics
Form of study:	Full-time
Supervisor:	Mgr. Michal Mičuda, Ph.D.

Thesis submitted on: . . . .

Univerzita Palackého
Přírodovědecká fakulta

Katedra experimentální fyziky

Elementární simulátor dynamiky jednoho qubitu

Bakalářská práce

Autor:	Marcel Drdla
Studijní program:	B1701 Fyzika
Studijní obor:	Aplikovaná fyzika
Forma studia:	Prezenční
Vedoucí:	Mgr. Michal Mičuda, Ph.D.
Práce odevzdána dne:	....

Abstract

This thesis deals with the implementation of arbitrary operation using a finite number of logic gates [1]. For example, repeating the rotation by an irrational angle which leads to the realization of any rotation along the given axis. That is presented through experimental implementation of an elementary quantum simulator for single qubit dynamics, which allows tests of long sequences of quantum operations. Such sequences result in operations with Hamiltonians that can correspond to sums or commutators of the Hamiltonians of the composing operations [2]. Each individual operation is realized as a sequence of state preparation, state transformation, and state estimation. The basic principle of the approach was demonstrated by chaining qubit operations with Hamiltonians σ_x and σ_y, in order to create Hamiltonian σ_z.

A major problem is the applicability of this approach. Various experimental imperfections accumulate with each repetition of these small transformations. For example, these imperfections include imperfect retardation of waveplates or their inaccurate rotation and unstable input optical power. MaxLik reconstruction method that is used may introduce additional systematic errors in each step.

In this thesis, we used polarized light to show how to use this approach to rotate a qubit along z-axis using x- and y-rotations only and studied the experimental challenges of the introduced method.

Keywords

`qubit, polarization, simulation, unitary operations`

Acknowledgments

I would like to thank Mgr. Michal Micuda, Ph.D. I appreciate his guidance, advice and patience. My thanks also goes to Mgr. Robert Stárek. for his beneficial consultations, hints and help with programming. I am grateful to other members of Department of Optics, especially to RNDr. Miroslav Ježek, Ph.D. and doc. Mgr. Petr Marek, Ph.D. for their supportive attitude. Last but not least I would like to thank my close family for their patience and support.

Marcel Drdla

Declaration

I hereby declare that I have written this Bachelor's Thesis "Elementary simulator of single qubit dynamics" and performed all the presented research and experimental tasks by myself, while being supervised by Mgr. Michal Mičuda, Ph.D. I also state that every resource used is properly cited. I agree with the Thesis being used for teaching purposes and being made available at the website of the Department of Optics.

Signed in Olomouc on ....

Marcel Drdla

Contents

Chapter 1

Introduction

1.1 Elementary simulator of single qubit dynamics

Quantum processing deals with the processing and coding of the quantum information [3]. Quantum information is coded in quantum bits - shortly qubits. Qubit differs from normal bits by not being limited to two logical states, logical states being 0 or 1. Qubits can exist in superpositions of these logical states, that is why we measure quantum states as probabilities.

Quantum processing requires the ability to transform these states arbitrarily [4]. A pure qubit could be described as a unit vector in three-dimensional space. An important category of single-qubit transformation is rotation. We would like to rotate the qubit arbitrarily, with a free choice of rotation axis and angle. A crucial principle of universality in quantum information processing [3] is that an arbitrary operation can be constructed from a finite set of operations.

In this thesis, we illustrate both theoretically and experimentally how to generate [5] a rotation along an axis from the other two non-commuting operators that generate rotatiom along other perpendicular. We will be proving the following formula

$$e^{iA\delta t}.e^{iB\delta t}.e^{-iA\delta t}.e^{-iB\delta t} = e^{[A,B]\delta t^2} + O(\delta t^3), \tag{1.1}$$

where A and B are the different operations, $[A,B] = AB - BA = C$ is commutator of operations used to create the desired operation C and $O(\delta t^3)$ is an error that affects to what extent is the sequence of operations identical to the operation we are replacing [2]. Element δt represents time. For our experiment $\delta t = \frac{\gamma}{N}$, where γ is arbitrary angle that parametrizes the rotation (for our experiment $\gamma = \frac{\pi}{2}$) and $\frac{\gamma}{N}$ then represents the amount of the operations A and B that will be executed.

Then equation (1.1) can be written as a sequence of repeating operations that make the desired operation.

$$\prod_{n=1}^{M}(e^{iA\frac{1}{N}t}.e^{iB\frac{1}{N}}.e^{-iA\frac{1}{N}}.e^{-iB\frac{1}{N}}) = e^{[A,B]\delta t^2.M} + O(\delta t^3.M), \tag{1.2}$$

where M is the number of repetitions of the small operation. Meaning that the larger M is, the more times the left part of the equation is written consecutively.

The error $O(\delta t^3)$ will then decrease with $N \to \infty$. The proof of the equality of equation (1.1) is done using Taylor series as

$$e^{iA\delta t}.e^{iB\delta t}.e^{-iA\delta t}.e^{-iB\delta t} = (1+iA\delta t).(1+iB\delta t).(1-iA\delta t).(1-iB\delta t) = \\ 1+(AB-BA)\delta t^2 + O(\delta t^3) = e^{(AB-BA)\delta t^2} + O(\delta t^3). \quad (1.3)$$

With the Taylor series we get the commutator of used operations and other elements with t^3, which are referred to as error $O(\delta t^3)$. This approximation is tested on one qubit. That means that the setup will be using just one laser beam and one detector with other components that are listed in Setup section.

In our experiment we are using Pauli matrices. Our goal is to realize z-rotations using x- and y- rotations. For example, on Bloch sphere the quantum state is defined with x, y, z coordinates. To change the coordinates of this state we can rotate x, y, z axes using Pauli matrices (σ-operations). Each Pauli matrix rotates one of these axes by π or 180° as mentioned before. Then the state is rotated around the z-axis, but for now we are assuming that this operation cannot be realized by using the Pauli matrix, which normally allows this movement (σ_z). We can refer to it as a forbidden operation. However, we can still rotate around x and y axis. If the state is rotated around x and y axes, meaning we use the other two Pauli matrices, the same transformation is done as if we would use the forbidden operation. So instead of using the forbidden operation, we are using a sequence of different operations, which we consider to be the same, with the correct choice of N.

In equation (1.1) operation $A = \sigma_x$ and $B = \sigma_y$. The commutator of these operations is equal to

$$[\sigma_x, \sigma_y] = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} - \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \\ \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix} - \begin{pmatrix} -i & 0 \\ 0 & i \end{pmatrix} = \begin{pmatrix} 2i & 0 \\ 0 & -2i \end{pmatrix} = 2i.\sigma_z. \quad (1.4)$$

Any operation can be realized using three retardation plates, namely two quarter wave plates (QWP) and one half wave plate(HWP) see Theory (2.16)-(2.17), in this order

$$U = QWP(\alpha).HWP(\beta).QWP(\delta), \quad (1.5)$$

where α, β, δ are rotation angles of corresponding retardation plates and U is desired operation. Experimental verification of this principle (1.1) can be realized by using these sigma operations in succession accordingly to the size of N. That is, however, impossible for large N due to the large number of components and space. Therefore, we will perform only one partial operation, measure the output and then set it as a new input for another measurement and we this will be performed M times. The larger the N is, the smaller the error $O(\delta t^3)$ gets [2]. However, in the experiment the imperfections accumulate. Hence our goal is to realize the forbidden operation and finding the optimal N so that the forbidden operation is as similar as possible to the one that will be created.

Chapter 2

Theory

2.1 Polarization

Let's assume a monochromatic wave with angular frequency ω, traveling in the direction of z-axis with velocity v [6]. Then the electric field vector lies in x-y plane and is described as

$$E(z,t) = Re\{A.e^{[i(\omega t - \frac{z}{v})]}\}, \tag{2.1}$$

where $A = A_x.\vec{e_x} + A_y.\vec{e_y}$ represents the electric field in x and y axes. If we substitute and expand the vector for electric field into cartesian coordinates we get the electric field operating in the x-axis and y-axis as

$$E_x(z,t) = A_x \cos[\omega t - \frac{z}{v} + \phi_1], \tag{2.2}$$

$$E_y(z,t) = A_y \cos[\omega t - \frac{z}{v} + \phi_2]. \tag{2.3}$$

If we add squared electric field in x-axis to squared electric field in y-axis we get the implicit form of ellipse equation

$$(\frac{E_x}{A_x})^2 + (\frac{E_y}{A_y})^2 - 2\frac{E_x}{A_x}\frac{E_y}{A_y}\cos(\delta) = \sin^2(\delta), \tag{2.4}$$

where $\delta = \phi_2 - \phi_1$ represents phase difference.

The Jones or Stokes formalism is used the most commonly for the description of polarization. We will use Jones formalism which is simpler but describes only pure states. The general form of Jones vector is then described as

$$J = \begin{pmatrix} A_x \\ A_y \end{pmatrix}. \tag{2.5}$$

$J^\dagger.J = 1$ applies to this vector, where † is Hermitian conjugation. Hermitian conjugation is equal to complex conjugation and transposition in any order.

Purity γ defines measure, giving information on how much is the polarization state J mixed [3]. If polarization state has $\gamma = 1$ then this is referred to as pure, meaning all of the light has the same polarization. We can also measure purity for density matrices defined as

$$\gamma = Tr(\rho^2) = Tr[(J.J^\dagger)^2], \tag{2.6}$$

where Tr is operation trace that gives us sum of elements on the main diagonal of square matrix.

The basic polarization states are described in Jones formalism in the following Table 2.1.

Polarization state	H	V	D	A	R	L
Vector representation	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$	$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{pmatrix}$	$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} \end{pmatrix}$	$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{-i}{\sqrt{2}} \end{pmatrix}$

Table 2.1: Demonstration of basic states and their expression in Jones formalism

Density matrix ρ [5], also called density operator, is a general description of the polarization state. Unlike polarization vectors, density matrices do not describe just pure states. Density matrix can describe a pure polarization state or a statistical mixture of them. If we know the Jones vector J that represents our polarization state, we can transform it into density matrix of size 2x2 as

$$\rho = J.J^\dagger \tag{2.7}$$

Fidelity is the magnitude of similarity between two density matrices [3]. It is a test of the probability that one state will be identified as the other. Generally, fidelity is defined as

$$F = [Tr[\sqrt{\sqrt{\rho}.\sigma.\sqrt{\rho}}]]^2. \tag{2.8}$$

However, for pure states or if at least one of the density matrices is pure ($\gamma = 1$), the equation (2.8) is simplified into

$$F = \frac{Tr[\rho.\sigma]}{Tr[\rho].Tr[\sigma]}. \tag{2.9}$$

To display polarization states we use Bloch sphere.

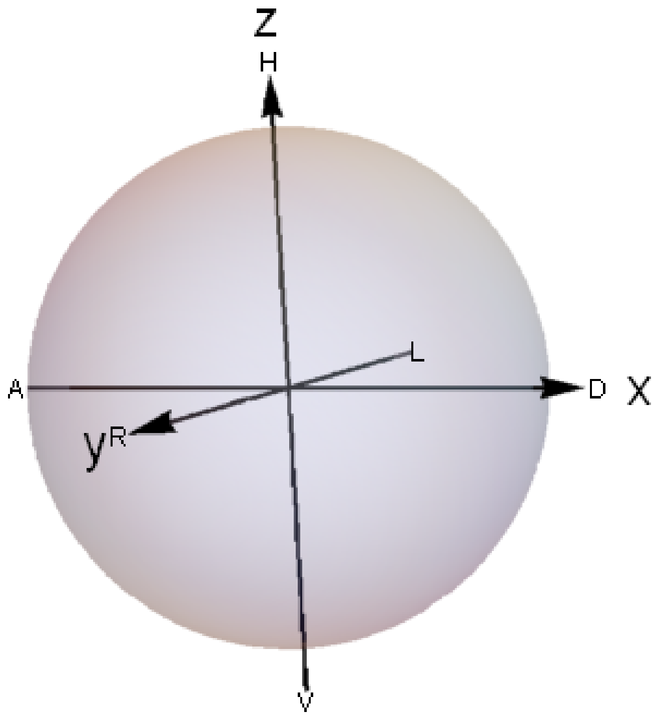


Figure 2.1: Bloch sphere with the basic Jones vectors shown

The Bloch sphere is a unit sphere in which there are three axes that correspond to certain basic polarization states (Table 2.1). All pure states are located on the surface of the Bloch sphere. All mixed states, states that have $\gamma < 1$, are located inside the Bloch sphere. If we want to display any polarization state, we use the Bloch vector $\vec{a} = (x, y, z)$. The coordinates for this vector are obtained from a density matrix ρ of a particular polarization state.

$$\begin{aligned} x &= \rho_{10} + \rho_{01} \\ y &= i.(\rho_{01} - \rho_{10}) \\ z &= \rho_{00} - \rho_{11} \end{aligned} \tag{2.10}$$

Hammer projection is used for better readability. This projection maps points on the surface of the sphere on its two-dimensional representation [7], see Figure 2.2. In further experiments we use this mapping to plot changes of polarization states.

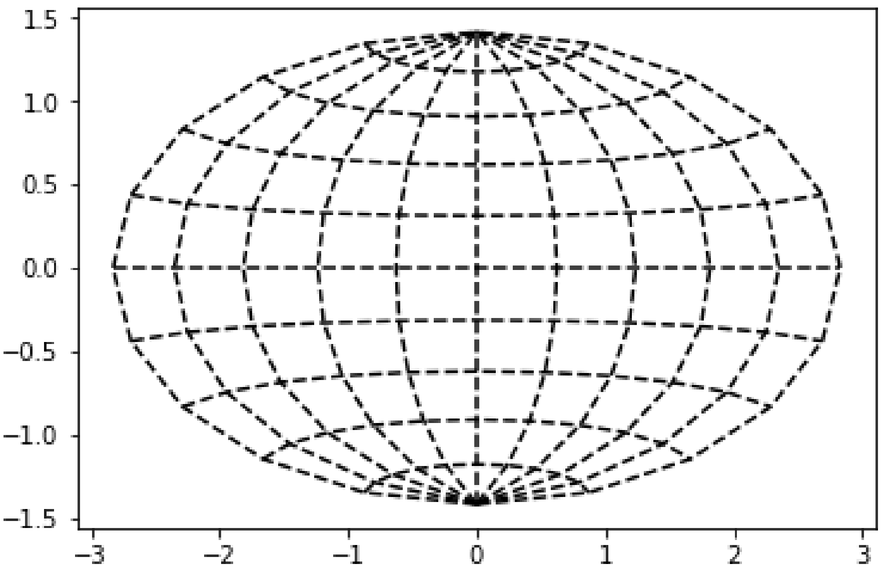


Figure 2.2: Bloch sphere in Hammer projection

Stokes formalism describes the polarization state of electromagnetic radiation with vector called Stokes vector. Stokes vector contains a set of four components. First component is purity (2.7) of the polarization state and the other three values are cartesian coordinates on the Bloch sphere. Therefore, it is the Bloch vector with the information about purity of particular polarization state. That means that we can describe any polarization state, even the impure polarization states. On the other hand, Jones formalism has a simpler form that is easier to work with.

2.2 Optical retarders

An optical retarder is an optical device that alters the polarization state of transmitted light wave and introduces a phase shift between the polarization components. Ideally, retarders do not polarize, cause change in the intensity, deviate or displace the light beam, they simply change their polarization form. Special cases of optical retarders are wave plates. Because we are assuming that all states are pure in our experiment, we can describe these states with Jones symbolism.

Wave plates are optical components that manipulate with the polarization of the transmitted light. General form of the transformation matrix for wave plates is

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{-i\delta} \end{pmatrix}, \tag{2.11}$$

A special case of wave plates that we will use are half- (HWP) and quarter-wave plates (QWP). These wave plates are usually made of birefringent material. This material adds respective delay (retardation) δ to the polarization. This delay depends on the thickness and rotation of the wave plate. For HWP, the delay will be $\delta = \frac{\pi}{2}$ and for QWP it will be $\delta = \frac{\pi}{4}$. If we substitute into the transformation matrix, we get matrices for HWP and QWP as

$$HWP = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{2.12}$$

$$QWP = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix}. \tag{2.13}$$

These matrices represent mentioned wave plates, when not rotated. We will need generally rotated wave plate. We define the rotation matrix as

$$Rot(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}. \tag{2.14}$$

Then we get the general rotated wave plates with the use of (2.12),(2.13) and (2.14) to get the matrix for generally rotated wave plates subsequently

$$T(\alpha) = Rot(-\alpha).T.Rot(\alpha). \tag{2.15}$$

And the matrices for HWP and QWP are defined as

$$HWP(\alpha) = \begin{pmatrix} \cos(2\alpha) & \sin(2\alpha) \\ \sin(2\alpha) & -\cos(2\alpha) \end{pmatrix}, \tag{2.16}$$

$$QWP(\alpha) = \begin{pmatrix} \cos^2(\alpha) - i\sin^2(\alpha) & \cos(\alpha)\sin(\alpha)(1+i) \\ \cos(\alpha)\sin(\alpha)(1+i) & \sin^2(\alpha) - i\cos^2(\alpha) \end{pmatrix}. \tag{2.17}$$

Another important optical element, that we will use, will be a polarizer. Polarizer is a polarization filter. That means that polarizer transmits only polarized light oscillating in one direction. The transmitted light depends on the rotation of the polarizer. The matrix form of linear polarizer, that transmits only horizontally polarized light, is the following

$$P = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}. \tag{2.18}$$

A matrix representation of a rotated polarizer is obtained similarly as in (2.15),

$$P(\alpha) = Rot(-\alpha).P.Rot(\alpha),$$

$$P(\alpha) = \begin{pmatrix} \cos^2(\alpha) & \cos(\alpha)\sin(\alpha) \\ \cos(\alpha)\sin(\alpha) & \sin^2(\alpha) \end{pmatrix}. \tag{2.19}$$

2.3 Pauli matrices

Pauli matrices are three 2x2 complex matrices that are Hermitian and unitary. We define them as σ-operations σ_x, σ_y and σ_z. If we plot our initial polarization state on Bloch sphere and apply one of the operations, two axes on the Bloch sphere will be mirror-shifted. That means that the last axis is rotated by π.

For example if we use σ_x operation on transmitted light and the light is polarized horizontally or vertically, then they will be rotated around the x-axis. Therefore, the horizontally polarized wave will be vertically polarized and vice versa. We define sigma operations as

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.20}$$

These operations can be realized with the use of wave plates. There are different ways to do these operations because any operation can be realized, as mentioned in Introduction (1.5), with the use of two $QWPs$ and one HWP. The simplest way to realize σ_x operation is using just one HWP rotated to 45° using formula (2.16). σ_y operation is realized as

$$\sigma_y = QWP(\frac{\pi}{2}).HWP(\frac{\pi}{4}).QWP(0). \tag{2.21}$$

2.4 Preparation and projection of polarized state

Generally, for the preparation and projection of polarized states we need three wave plates, two $QWPs$ and one HWP (1.5). In the experiment we use setup that ensures that the input state is horizontally polarized light. Then we use one QWP and one HWP in this order which allows us to prepare any arbitrary polarized state. Subsequent formula represents preparation of the arbitrary polarized output

$$OUT = HWP(\beta).QWP(\alpha).IN, \tag{2.22}$$

where OUT is polarization we want and IN is horizontally linear polarization we have coming to the setup. Angles α and β are found by assuming that the state IN is horizontal and then the plates should transform this state into all basic polarization states Table 2.1.

The reverse order of the plates is used for projection

$$OUT = QWP(\beta).HWP(\alpha).IN, \tag{2.23}$$

where OUT is the state that we are projecting to and IN is the transformed state from the setup. For projection, the angles α and β are found by assuming that the state IN is one of the basic states Table 2.1. Then for each of these states the angles are found so that the OUT state is horizontally polarized.

To measure a projection of a state A to a state B means to measure the probability $|A^{\dagger}.B|^2$ [5]. To reveal an unknown polarization state, we use two wave plates to set the projection state. The unknown state is projected into all of the Jones vectors from Table 2.1 using angles for wave plates from Table 2.2. The projected state then goes through the polarizer and the intensity is measured on detector. Then we should have six different intensities where each corresponds to one of the Jones vectors. From this we can see which states are mostly contained in the unknown polarization. If the projected state intensity is biggest for one of the Jones vectors and equal to zero in the opposite to the biggest one, we can conclude that the unknown state is the Jones vector with the biggest intensity.

The data coming from our projection are used for a reconstruction of the unknown state. One of the reconstruction methods of the polarization analysis is used to obtain the state density matrix ρ.

	Preparation		**Projection**	
State	**QWP**(α)	**HWP**(β)	**HWP**(β)	**QWP**(α)
H	0°	0°	0°	0°
V	0°	45°	45°	0°
D	0°	22.5°	22.5°	0°
A	0°	−22.5°	−22.5°	0°
R	−45°	−22.5°	22.5°	45°
L	45°	22.5°	−22.5°	−45°

Table 2.2: Angles for wave plates used for preparation and detection in the experiment

2.5 State reconstruction

The reconstruction method we are using is called Maximum Likelihood estimation [8] known as MaxLik to reconstruct state density matrix ρ (2.7) from acquired data in the projection process. The advantage of this method is that it will always have a physical output. Unfortunately, every reconstruction method that outputs physical results suffers from systematic errors [9]. For example, purity and fidelity estimation is biased. This effect is stronger when the tomography is limited to a small number of copies. This fact is not very relevant in our case as in the experiment because we are working with states with the purity values close to one, and we are working with a strong laser beam.

For the MaxLik method, the unknown state is projected into basic polarization states shown in Table 2.1. This then gives us a list of six values $data = [H, V, D, A, R, L]$, which are the input data to this method. Another input for the MaxLik method is a list of projectors corresponding to the provided data. From these inputs, a density matrix of the respective unknown state is reconstructed. The reconstruction algorithm iteratively searches for the quantum state that reproduces the recorded data with the largest quasi-probability (likelihood).

2.6 Similarity of unitary operations

To compare unitary operations, a conversion from a 2x2 matrix to a 4x4 process matrix χ, called Choi – Jamiołkowski isomorphism [10, 11] is used. This process matrix χ is used to describe a unitary operation U and their relation is

$$\chi = (I \otimes U).(|00\rangle + |11\rangle).(\langle 00| + \langle 11|).(I \otimes U)^\dagger, \tag{2.24}$$

where I is identity matrix, $\otimes$ is the Kronecker product and

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \langle 00| = |00\rangle^\dagger, \langle 11| = |11\rangle^\dagger. \tag{2.25}$$

This conversion is used to compare two unitary operations with infidelity INF, which gives information about the difference between the two operations.

$$INF = 1 - F, \tag{2.26}$$

where F is fidelity (2.9) measured for the two unitary operations. This comparison has the advantage that it can be used to compare even imperfect operations. Figure 2.3 shows a comparison of unitary operations from equation (1.1), where as mentioned before $A = \sigma_x$ and $B = \sigma_y$. The operations from the left and right sides are compared using infidelity, which then shows us the error $O(\delta t^3)$ that decreases with increasing N.

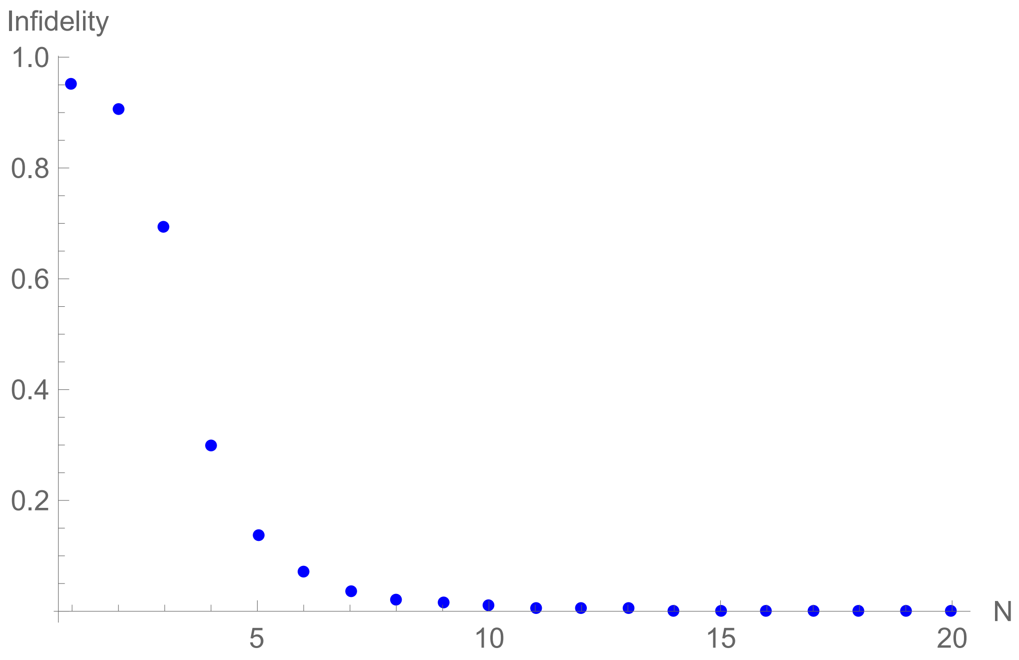


Figure 2.3: Process infidelity between left and right sides of equation (1.1)

We parametrize a unitary operation with parameters θ, ϕ_1, and ϕ_2 in a following way

$$U(\phi_1, \phi_2, \theta) = \begin{pmatrix} \alpha & -\overline{\beta} \\ \beta & \overline{\alpha} \end{pmatrix} \tag{2.27}$$

where $\alpha = e^{i\phi_1}. \cos(\theta)$, $\beta = e^{i\phi_2}. \sin(\theta)$ and $\overline{\alpha}$ means complex conjugation. This parametrization is called the special unitary (SU2) matrix [5].

We would like to correct the experimental operation χ_{exp} to achieve better fidelity with desired operation χ_{the}. To do so, we apply a unitary operation U_{θ,ϕ_1,ϕ_2} on the experimental process matrix to obtain the corrected process matrix.

$$\chi_{\text{cor}} = (1 \otimes U_{\theta,\phi_1,\phi_2})\, \chi_{\text{exp}}\, (1 \otimes U_{\theta,\phi_1,\phi_2})^{\dagger} . \tag{2.28}$$

We search for parameters θ, ϕ_1, ϕ_2 that maximize the fidelity $F(\chi_{\text{cor}}, \chi_{\text{the}})$. Resulting parameters describe the best corrective unitary operation that we can use to compensate for errors in the experiment.

Chapter 3

Experiment

In all experiments, we assume that the input state is horizontally polarized, which is then transformed into a state with which we perform the required measurements using a preparation block. Therefore, our task will be to prepare a diagonal state, which if we applied the right side of the equation (1.1) to it, it should be transformed into an antidiagonal state. So we will try to carry out this transformation using the left side of the equation (1.1) which we apply to the input state according to the chosen N so that we get as close as possible to this desired antidiagonal system. The similarity between the wanted state and the transformed is measured with fidelity (2.9).

3.1 Simulation

The simulation is carried in the same way as the experiment will run. The whole simulation is firstly done in Wolfram Mathematica. This program was chosen because of its flexibility in calculation of the lines we want to do and its simple programming language. For further calculations, the script is written in Python. Partial operations (1.1) that are used to achieve the operation $e^{i\sigma_z \cdot \delta}$ (where δ is an arbitrary angle that for us is $\frac{\pi}{2}$) are performed in the experiment.

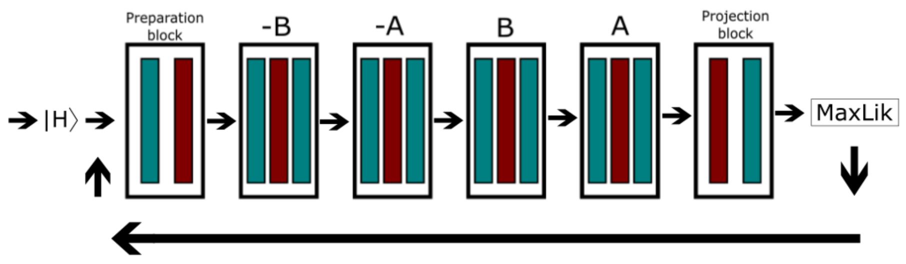


Figure 3.1: The visualization of the setup for simulation, where blue blocks represent QWP and red blocks represent HWP.

The simulation is divided into three basic parts. The first is the input state preparation, the second is the input state passage through the setup, basic state projection and acquisition of the density matrix with MaxLik (see Theory 2.5) and finally the third is the searching for the angles to set the output as the new input state using density matrix.

The input state preparation is realized with two wave plates (2.22) with the Input state being horizontally linear polarized light that is coming to the setup.

The next block is constructed according to the left side of the equation (1.1), where $A = \sigma_x$ and $B = \sigma_y$. According to the chosen N we use only a part of these operations. These partial operations can be written as

$$e^{i\sigma_{x,y}\frac{\pi}{2}\frac{1}{N}} = I.\cos(\frac{\pi}{2N}) + i\sigma_{x,y}.\sin(\frac{\pi}{2N}), \tag{3.1}$$

where I is an identity matrix and besides adding the operation, we also add angle $\frac{\pi}{2}$ that represents the full execution of any σ-operation.

With this equation, we try to find a relation between the settings (rotation angles) of the wave plates that change with different N. These partial operations are realized by three wave plates, two $QWPs$ and one HWP. Different angles for the wave plates are tested experimentally and compared with the matrix that originates from the equation (1.1) until an analytical relation – dependence and a change with N - is found for the angles. According to chosen N, these operations can be written as

$$e^{i\sigma_x\frac{\pi}{2N}} = QWP(\frac{\pi}{2}).HWP(\frac{(2N+1).\pi}{4N}).QWP(\frac{\pi}{2}), \tag{3.2}$$

$$e^{i\sigma_y\frac{\pi}{2N}} = QWP(0).HWP(\frac{\pi}{4N}).QWP(\frac{\pi}{2N}), \tag{3.3}$$

$$e^{-i\sigma_x\frac{\pi}{2N}} = QWP(\frac{\pi}{2}).HWP(\frac{(2N-1).\pi}{4N}).QWP(\frac{\pi}{2}), \tag{3.4}$$

$$e^{-i\sigma_y\frac{\pi}{2N}} = QWP(\frac{\pi}{2N}).HWP(\frac{\pi}{4N}).QWP(0). \tag{3.5}$$

With these assesed settings for wave plates, the left side of the equation (1.1) can be written as

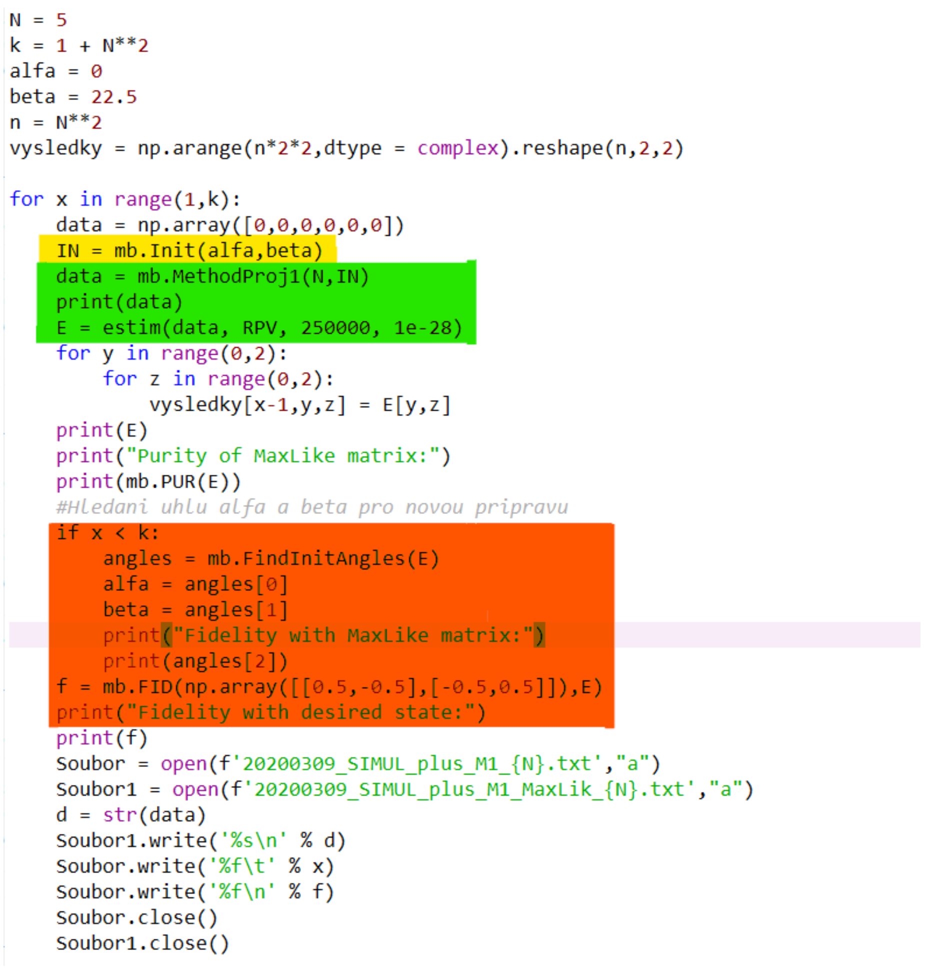
$$\begin{aligned} &e^{i\sigma_x\frac{\pi}{2N}}.e^{i\sigma_y\frac{\pi}{2N}}.e^{-i\sigma_x\frac{\pi}{2N}}.e^{-i\sigma_y\frac{\pi}{2N}} = \\ &QWP(\frac{\pi}{2}).HWP(\frac{(2N+1).\pi}{4N}).QWP(\frac{\pi}{2}).QWP(0).HWP(\frac{\pi}{4N}).QWP(\frac{\pi}{2N}). \\ &QWP(\frac{\pi}{2}).HWP(\frac{(2N-1).\pi}{4N}).QWP(\frac{\pi}{2}).QWP(\frac{\pi}{2N}).HWP(\frac{\pi}{4N}).QWP(0). \end{aligned} \tag{3.6}$$

With the knowledge of proper settings for any given N, we let the input state from the first block go through the setup (3.6). The output state given is then projected into the basic Jones vectors (Table 2.1) using QWP and HWP (2.23) with the angles shown in Table 2.2. For the resulting intensities from the projection, the MaxLik method is used, which gives us a density matrix ρ of the output state. In addition, we calculate purity (2.7) (PUR in script) for this matrix.

The last block of the simulation uses the density matrix ρ' to find such angles for another cycle of the simulation to change the input state of the new cycle as the output of the current cycle. We are trying to find a density matrix ρ' that is prepared with the following formula

$$\rho' = [HWP(i).QWP(q).H].[HWP(i).QWP(q).H]^{\dagger}, \tag{3.7}$$

where i and q are the angles that we are trying to find, and H is the horizontal polarization state from Table 1. Then we use two For cycles that run for both unknown angles from $\frac{-\pi}{2}$ to $\frac{\pi}{2}$ with step of 0.005 radians. We create two lists, where one is filled with angles used to create density matrices (3.7) and the second one is filled with these created matrices. First, we take the second list and make another list filled with values of fidelity between our created matrices and the matrix from the second block of our simulation. Then we find the maximal value of fidelity that should be close to one and find the position of this maximal value. This position is the same for the other list from which the density matrix and angles used for creating it are obtained. After that, the angles are set as the new initial angles for the preparation and the cycle starts from the beginning. This cycle is repeated $M = N^2$ times (1.2) and the last block that finds the new initial angles will not be performed.



```python
N = 5
k = 1 + N**2
alfa = 0
beta = 22.5
n = N**2
vysledky = np.arange(n*2*2,dtype = complex).reshape(n,2,2)

for x in range(1,k):
    data = np.array([0,0,0,0,0,0])
    IN = mb.Init(alfa,beta)
    data = mb.MethodProj1(N,IN)
    print(data)
    E = estim(data, RPV, 250000, 1e-28)
    for y in range(0,2):
        for z in range(0,2):
            vysledky[x-1,y,z] = E[y,z]
    print(E)
    print("Purity of MaxLike matrix:")
    print(mb.PUR(E))
    #Hledani uhlu alfa a beta pro novou pripravu
    if x < k:
        angles = mb.FindInitAngles(E)
        alfa = angles[0]
        beta = angles[1]
        print("Fidelity with MaxLike matrix:")
        print(angles[2])
    f = mb.FID(np.array([[0.5,-0.5],[-0.5,0.5]]),E)
    print("Fidelity with desired state:")
    print(f)
    Soubor = open(f'20200309_SIMUL_plus_M1_{N}.txt',"a")
    Soubor1 = open(f'20200309_SIMUL_plus_M1_MaxLik_{N}.txt',"a")
    d = str(data)
    Soubor1.write('%s\n' % d)
    Soubor.write('%f\t' % x)
    Soubor.write('%f\n' % f)
    Soubor.close()
    Soubor1.close()
```

Figure 3.2: The simulation script, where yellow represents the initial state preparation, green second block and orange the finding of new starting angles

The simulation is carried for several different N and reflects the state that would result from using the right side of the equation (1.1) only. The errors in Figure 3.3 are plotted using the fidelity (2.9).

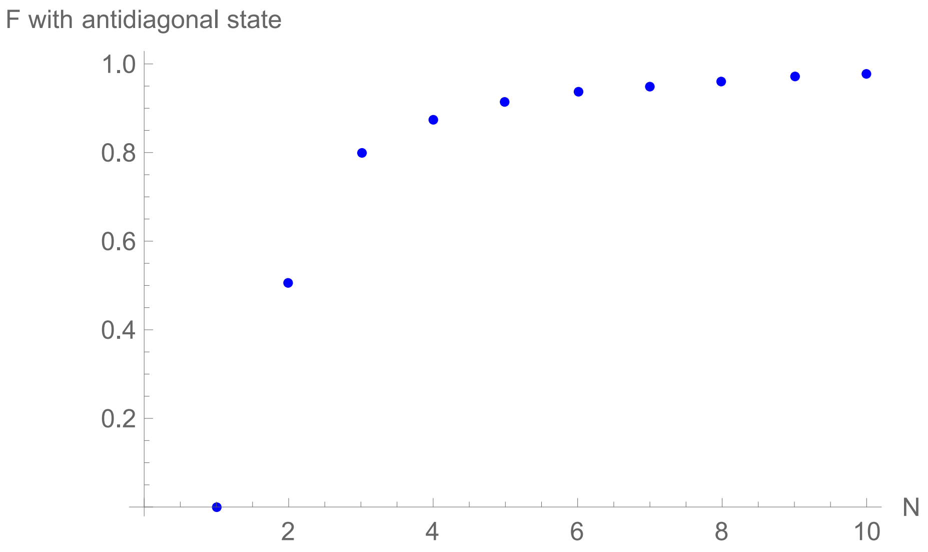


Figure 3.3: Fidelity with desired antidiagonal state

3.2 Setup

The core of our experimental setup consists of polarizers, $QWPs$, $HWPs$ and detector assembled in the following Figure 3.4.

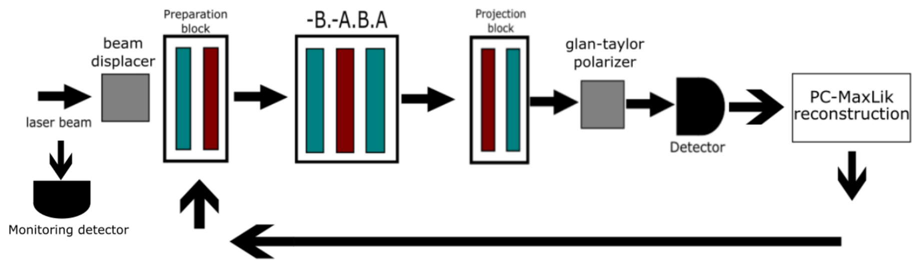


Figure 3.4: The visual representation of used setup

As a polarizer we are using a calcite polarizing beam displacer that splits the beam into two polarizations, horizontal and vertical, where the horizontal is deflected when passing through the beam splitter and straightened again at the exit of the beam splitter so that there are two parallel rays. The deflected beam with horizontal polarization is stopped with a beam stopper, therefore there is only one beam with vertical polarization. In theory, we assumed horizontally-polarized state on input, while in the actual experiment, we used a vertically-polarized state as an input. It is straightforward to take this fact into account. We only needed to use the correct IN vector in relation (2.22).

Glan-Taylor polarizer is placed before the detector. The source used is a laser diode with a wavelength of 810 nm from which light is output by means of an optical fiber. The single-mode optical fiber guides the light to the breadboard with the experimental setup. The fiber passes through the mechanical fiber polarization controller at the breadboard, which influences the fiber's polarization. The light is decoupled into free space using a collimator with an 11-mm effective focal length. The detector is a free-space PIN diode, which translates the captured light to a current which is then measured using an ammeter. Using a 50/50 laser splitter, one signal is fed to the setup and the other to a monitoring detector for measuring laser's stability. This second beam is measured to see how the setup influenced the laser's stability.

Motorized $QWPs$ and $HWPs$ are used for the preparation and projection. For the block representing the wanted operation, plates in a manual rotation mounts are used. Motorized plates can be used here as well, however, it is not necessary as the plates are set at the beginning of each experiment and are no longer moved afterward.

3.3 Preparing the experiment

At first, we need to prepare the setup, learn how the motorized rotations work and write script for the experiment. The first setup is therefore intended for the preparation of the mentioned and there would not be any additional filters – only $QWPs$, $HWPs$, beam splitters and detector.

Instructions on how to build and find the starting angles for retardation plates can be found in Annexes 5.1. The setup is built (preparation a projection blocks only) with manual $QWPs$ and $HWPs$. The starting angles of the wave plates are found and the testing is performed to assess if the setup is working as expected. The testing is based on verifying the validity of the angles given in Table 2.2 by means of preparation and projection of basic Jones states. Then intensities measured by the detector are used to perform MaxLik method and to reconstruct a density matrix. The basic density matrix and state matrix are prepared to find the correct settings for the MaxLik method. Then the manual retardation plates are replaced with motorized ones and we find their starting angles. If we want to write the script for our experiment first, we need to know how to move the motorized rotations. How commands are sent and connected to the laptop is explained in the manual [12] for motorized rotations. Commands that are used in the experiment are available in the following Table 3.1.

Command	Description	Output
xxPAXX	absolute move to value of XX	returns the target position value
xxPTXX	motion time in seconds to move for the value of XX	returns the time in seconds
xxTP	gets current position of xx rotation	return value of the current position
xxMMXX	changes controller's state for XX=0 READY to DISABLE, XX=1 DISABLE to READY	returns the current state of controller
xxOR	starts execution of HOME search	—

Table 3.1: Commands in the script used to control rotations where xx-number of the rotations controller, XX-optional or required value

With the knowledge of these commands we can write simple program that sets input state, gets intensities measured on the detector from projections, saves them and then performs MaxLik method and prints reconstructed matrix with its purity.

When the script is ready, a laser stability test can be performed. This is done simply by setting an initial state that is projected into another state for which we get half the intensity of the laser. We want a different initial state than a horizontal one because that is the polarization from the laser. Therefore, we want to transform it so that we know if the retardation plates are not affecting the polarization state in time. Our initial state is diagonally polarized. The stability of the laser is shown in Figure 3.5.

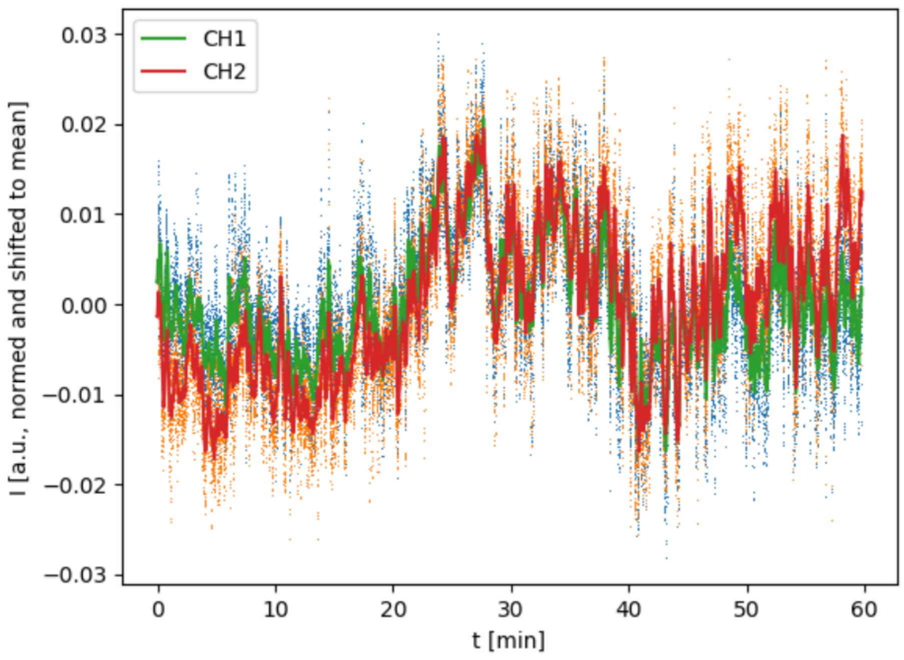
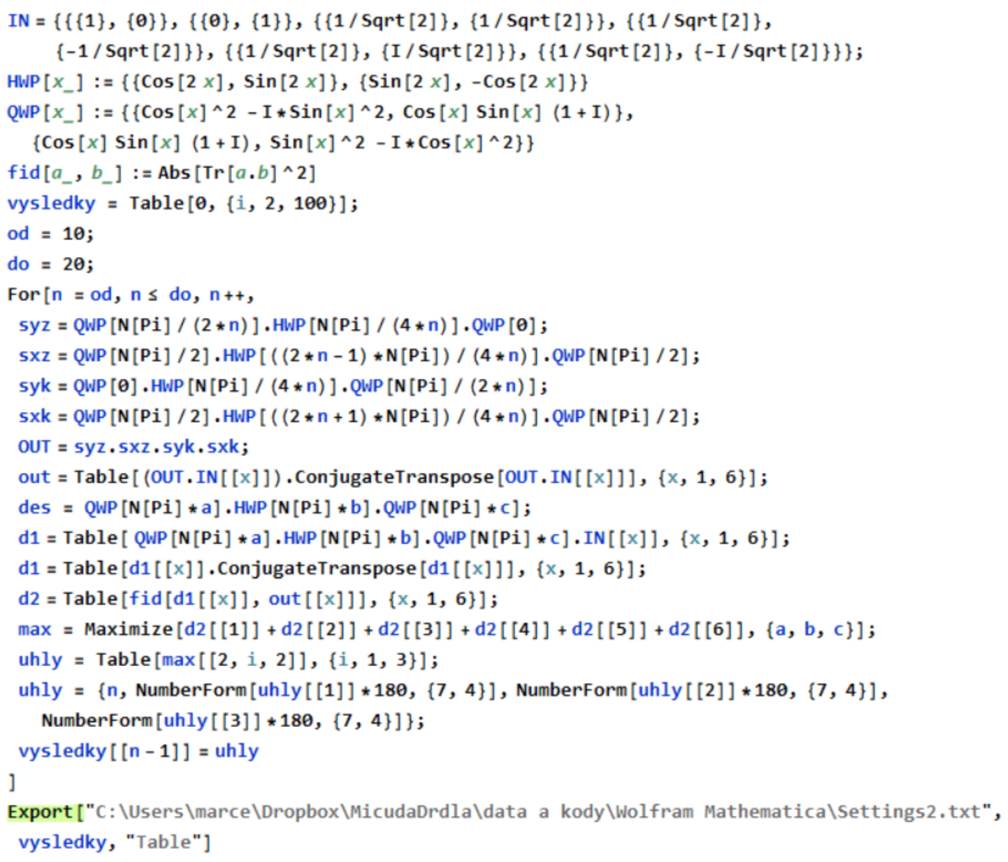


Figure 3.5: Laser stability test in time before the final measurements

Because there is not enough physical space for all twelve retardation plates that will be placed in the middle of our setup, we need to come up with a different solution. As mentioned in Theory, a section of any unitary operation can be realized with just two $QWPs$ and one HWP between them. So instead of placing twelve plates we use only three.

The problem is that we know how the rotation angles change for different N and that the change is linear (3.2)-(3.5), but we do not know how the rotation angles change for these three plates only. If we try to find a simple rule for these three plates, we find that the angles change non-trivially. This is solved by another script written in Wolfram, that finds appropriate angles for specified N. Subsequently, for the given N, it creates a matrix of multiplied twelve plates (3.6) and creates a second matrix of three plates that contain the variables α, β, γ. Then these matrices are multiplied with six basic Jones vectors and density matrices are formed from them. Now we have twelve density matrices, six of which are created from the known rule for twelve plates and six of them still incomplete because they contain the variables α, β and γ.

Using fidelity and the prescribed *Maximize* function, we perform the sum of six fidelities, which we maximize using this function so that this sum is equal to six. *Maximize* function finds suitable variables (α, β, γ) that ensure the maximum fidelity. Now the script just converts these found angles from radians to degrees and prints them out.



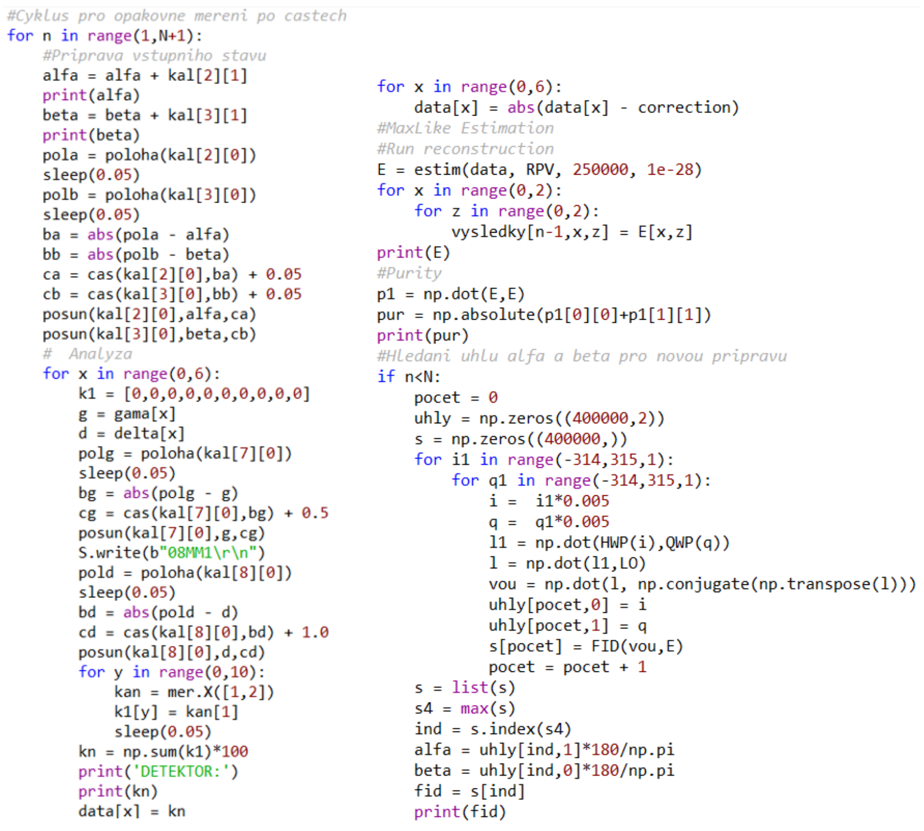
```
IN = {{{1}, {0}}, {{0}, {1}}, {{1/Sqrt[2]}, {1/Sqrt[2]}}, {{1/Sqrt[2]},
    {-1/Sqrt[2]}}, {{1/Sqrt[2]}, {I/Sqrt[2]}}, {{1/Sqrt[2]}, {-I/Sqrt[2]}}};
HWP[x_] := {{Cos[2 x], Sin[2 x]}, {Sin[2 x], -Cos[2 x]}}
QWP[x_] := {{Cos[x]^2 - I*Sin[x]^2, Cos[x] Sin[x] (1+I)},
  {Cos[x] Sin[x] (1+I), Sin[x]^2 - I*Cos[x]^2}}
fid[a_, b_] := Abs[Tr[a.b]^2]
vysledky = Table[0, {i, 2, 100}];
od = 10;
do = 20;
For[n = od, n ≤ do, n++,
 syz = QWP[N[Pi] / (2*n)].HWP[N[Pi] / (4*n)].QWP[0];
 sxz = QWP[N[Pi] / 2].HWP[((2*n - 1) *N[Pi]) / (4*n)].QWP[N[Pi] / 2];
 syk = QWP[0].HWP[N[Pi] / (4*n)].QWP[N[Pi] / (2*n)];
 sxk = QWP[N[Pi] / 2].HWP[((2*n + 1) *N[Pi]) / (4*n)].QWP[N[Pi] / 2];
 OUT = syz.sxz.syk.sxk;
 out = Table[(OUT.IN[[x]]).ConjugateTranspose[OUT.IN[[x]]], {x, 1, 6}];
 des = QWP[N[Pi] *a].HWP[N[Pi] *b].QWP[N[Pi] *c];
 d1 = Table[ QWP[N[Pi] *a].HWP[N[Pi] *b].QWP[N[Pi] *c].IN[[x]], {x, 1, 6}];
 d1 = Table[d1[[x]].ConjugateTranspose[d1[[x]]], {x, 1, 6}];
 d2 = Table[fid[d1[[x]], out[[x]]], {x, 1, 6}];
 max = Maximize[d2[[1]] + d2[[2]] + d2[[3]] + d2[[4]] + d2[[5]] + d2[[6]], {a, b, c}];
 uhly = Table[max[[2, i, 2]], {i, 1, 3}];
 uhly = {n, NumberForm[uhly[[1]] *180, {7, 4}], NumberForm[uhly[[2]] *180, {7, 4}],
   NumberForm[uhly[[3]] *180, {7, 4}]};
 vysledky[[n - 1]] = uhly
]
Export["C:\Users\marce\Dropbox\MicudaDrdla\data a kody\Wolfram Mathematica\Settings2.txt",
 vysledky, "Table"]
```

Figure 3.6: The Wolfram Mathematica script that finds angles for the three wave plates in Figure 6

With the fact that we know how to rotate the plates for any N, the whole setup is built again, this time with our three plates that represent the desired partial operations with additional filters added. The final script that consists of the same parts described in Simulation, however with certain differences, is written. There is no block that would simulate the passage of our initial state through the setup. Before every experiment we need to manually set the three plates in the middle to the found angles with the script from Wolfram Mathematica.

First the script sets the preparation block to the defined angles, then makes all projections and saves the data from the detector into a list. With this list the density matrix can be reconstructed using the MaxLik method. Then the purity is printed out of this reconstructed matrix to ensure its value is close to one. The last part of the cycle is finding of new initial angles and saving of the measured data from detector, reconstructed matrix, and its purity. The finding of the new angles is described in Simulation in the third block.

The only thing that we need to always assign is the first set of angles that are almost always the angles for diagonal polarization, the name of the file in which the data will be saved and N for which the measurements are performed.



```
#Cyklus pro opakovne mereni po castech
for n in range(1,N+1):
    #Priprava vstupniho stavu
    alfa = alfa + kal[2][1]
    print(alfa)
    beta = beta + kal[3][1]
    print(beta)
    pola = poloha(kal[2][0])
    sleep(0.05)
    polb = poloha(kal[3][0])
    sleep(0.05)
    ba = abs(pola - alfa)
    bb = abs(polb - beta)
    ca = cas(kal[2][0],ba) + 0.05
    cb = cas(kal[3][0],bb) + 0.05
    posun(kal[2][0],alfa,ca)
    posun(kal[3][0],beta,cb)
    #  Analyza
    for x in range(0,6):
        k1 = [0,0,0,0,0,0,0,0,0,0]
        g = gama[x]
        d = delta[x]
        polg = poloha(kal[7][0])
        sleep(0.05)
        bg = abs(polg - g)
        cg = cas(kal[7][0],bg) + 0.5
        posun(kal[7][0],g,cg)
        S.write(b"08MM1\r\n")
        pold = poloha(kal[8][0])
        sleep(0.05)
        bd = abs(pold - d)
        cd = cas(kal[8][0],bd) + 1.0
        posun(kal[8][0],d,cd)
        for y in range(0,10):
            kan = mer.X([1,2])
            k1[y] = kan[1]
            sleep(0.05)
        kn = np.sum(k1)*100
        print('DETEKTOR:')
        print(kn)
        data[x] = kn
```

```
for x in range(0,6):
    data[x] = abs(data[x] - correction)
#MaxLike Estimation
#Run reconstruction
E = estim(data, RPV, 250000, 1e-28)
for x in range(0,2):
    for z in range(0,2):
        vysledky[n-1,x,z] = E[x,z]
print(E)
#Purity
p1 = np.dot(E,E)
pur = np.absolute(p1[0][0]+p1[1][1])
print(pur)
#Hledani uhlu alfa a beta pro novou pripravu
if n<N:
    pocet = 0
    uhly = np.zeros((400000,2))
    s = np.zeros((400000,))
    for i1 in range(-314,315,1):
        for q1 in range(-314,315,1):
            i =  i1*0.005
            q =  q1*0.005
            l1 = np.dot(HWP(i),QWP(q))
            l = np.dot(l1,L0)
            vou = np.dot(l, np.conjugate(np.transpose(l)))
            uhly[pocet,0] = i
            uhly[pocet,1] = q
            s[pocet] = FID(vou,E)
            pocet = pocet + 1
    s = list(s)
    s4 = max(s)
    ind = s.index(s4)
    alfa = uhly[ind,1]*180/np.pi
    beta = uhly[ind,0]*180/np.pi
    fid = s[ind]
    print(fid)
```

Figure 3.7: The script used for measurements

3.4 Simulation of the error

The first thing that is tested is the error $O(\delta t^3)$ from equation (1.1) in Python. The test consists of comparing the left and right side of this equation as shown in Figure 2.3 and the found formula (3.6) with the right side of the equation (1.1) with infidelity (2.27) as shown in Figure 3.8. These Figures are compared so that the infidelities we draw in them are plotted in one graph where they perfectly overlapped. If we look directly at these infidelities, we find that they have exactly the same values for the same N. The corresponding script can be found in Annexes see Figures 5.12 and 5.13. This infidelity is plotted using Wolfram Mathematica.

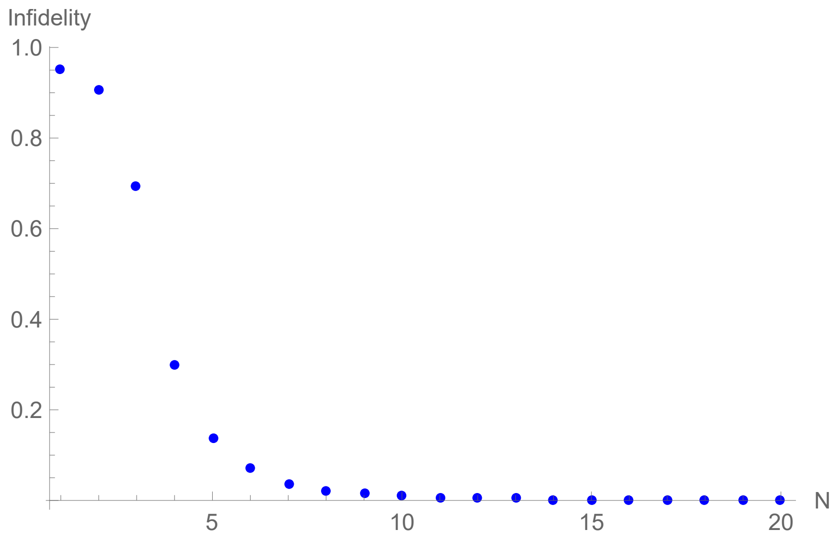


Figure 3.8: The error when comparing right side of the equation (1.1) with the found formula (3.6)

The data from these two tests are identical, meaning our found formula directly corresponds to the left side of the equation. The main problem is that we are comparing unitary operations. These operations cannot be compared as matrices 2x2 as they are not density matrices. This problem has two solutions. First one is applying these operations on some polarization state and then the output vector is converted to density matrix and the two density matrices are compared with fidelity. However, this solution has one flaw and that is the chosen vector that can be eigenstate of this unitary operation. To be sure that this does not happen, it is recommended to use more polarization states and compare all of these density matrices together. Or the unitary operation can be converted from size 2x2 to 4x4 using Choi–Jamiołkowski isomorphism (2.24). This formula is better for comparison of two unitary operations, because it does not suffer from the previously discussed drawback of the first method.

3.5 Data from the experiment

The first measurement is done for different number of cycles from 2 to 100 for $N = 10$ and the state in each step is compared to antidiagonal state with fidelity (2.9). The initial state that we prepared was diagonaly polarized. These measurements are done to test out the setup and to find any errors in the script. With these measurements we found out that some of our projection angles have been swapped, specifically the D, A states were reversed with R, L see Table 2.1.

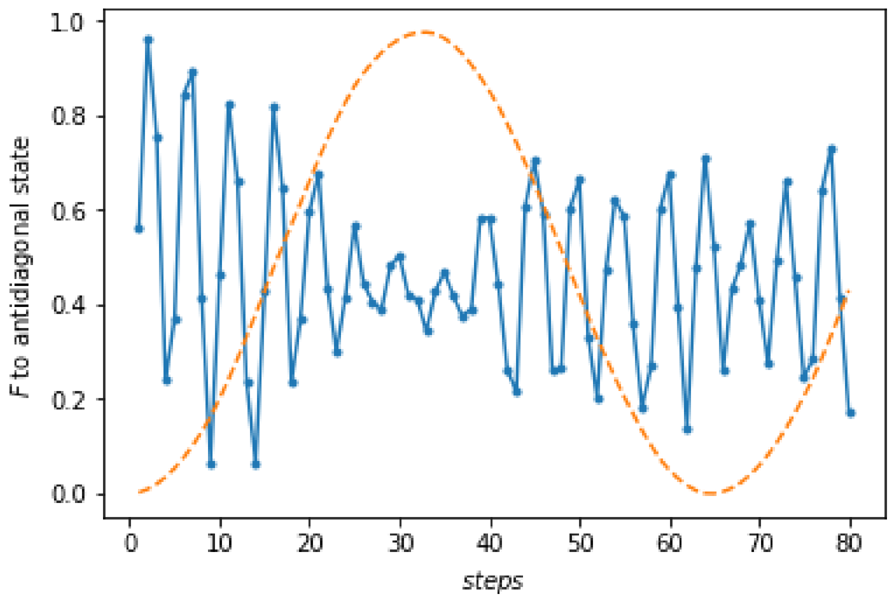


Figure 3.9: Fidelity to antidiagonal state with cycle of 80 steps and $N = 10$, blue dots represents measured data and orange dashed line theoretical data

These first measurements show that the change does not correspond with the theory. That is why the error needs to be found. For this purpose the laser stability is measured. The stability is measured without the retardation plates and then gradually the plates are added until the setup is fully built again. Each addition to the setup is measured separately. Therefore, each measurement represents the development of the laser stability in time and additionally the transformation of the intensity done by the plates.

The laser's stability in time is normalized and if the intensity of the laser ranges from a mean value to 0.5% then the laser is considered stable. Another stability test is done before the final measurements, see Figure 3.5. The rest of the stability measurements can be found in Annexes.

As shown in Figure 3.5, the laser is sufficiently stable and therefore it is excluded as being an influence on experiments. Then before checking if the retardation plates are influencing the polarization, unitary correction matrix U_{corr} is implemented to compensate the error. This unitary correction is found using process tomography. The reconstructed density matrix is compared with theoretical density matrix of the process using fidelity (2.9) that we maximize. The theoretical matrix is made using the special unitary group matrix SU2 (2.27), which gives density matrix of size 2x2 and this matrix is then transformed to size 4x4 with formula (2.24). After the parameters for SU2 matrix are found, this unitary correction matrix is used to transform the reconstructed density matrix as seen in following equation

$$\rho_{prep} = U_{corr}.\rho_{meas}.U^{\dagger}_{corr}, \tag{3.8}$$

where ρ_{meas} is the measured density matrix and ρ_{prep} is the corrected density matrix.

Following Figures 3.10-3.11 shows the evolution of identity, that means that the three middle plates should not transform the polarization in any way. The input state is diagonally polarized.

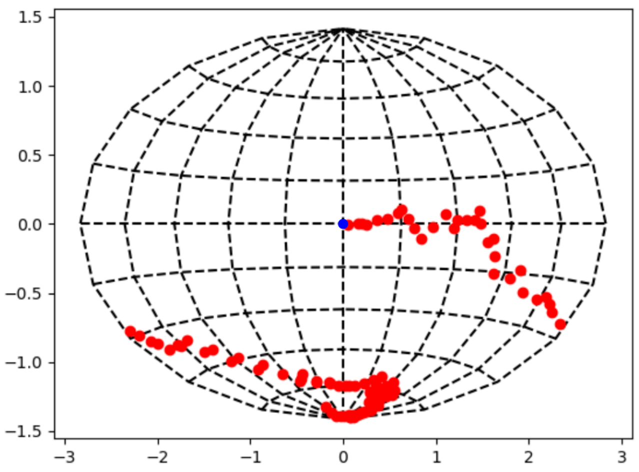


Figure 3.10: The process of identity evolution without the correction matrix, where blue point represents theoretical evolution

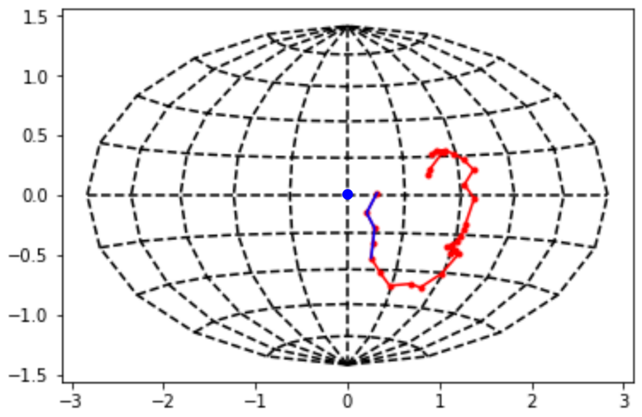


Figure 3.11: The process of identity evolution when using a correction matrix, where blue point represents theoretical evolution

In Figure 3.11 the first five cycles are connected with blue line. As we can see the state moves towards vertical polarization instead of antidiagonal. The error is therefore not fully compensated and therefore we look for it further.

The next test is using σ_x operation. For this operation the initial state is horizontally polarized, so this operation should mirror shift it to become vertically polarized and so on.

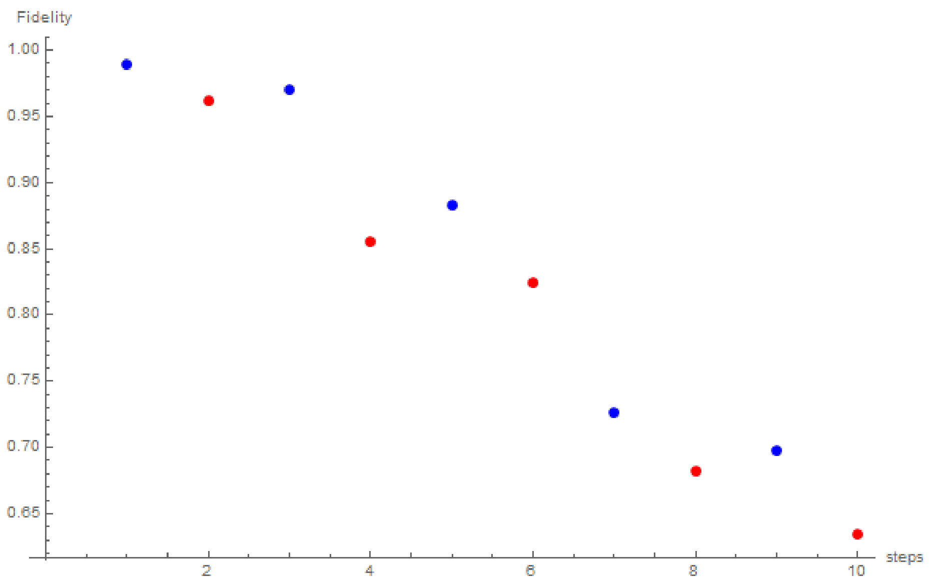


Figure 3.12: Fidelity to corresponding states for σ_x operation, where blue is fidelity with vertical state and red is fidelity with horizontal state

With the correction matrix another identity evolution is measured, this time with diagonal initial state. The following Figure 3.13 shows the best agreement with the theory we have achieved so far.

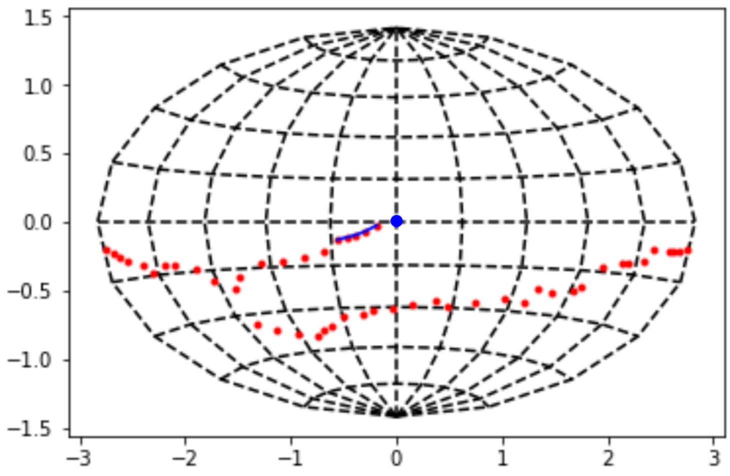


Figure 3.13: The process of identity evolution with better correction matrix, where blue point represents theoretical evolution

This correction helps, but the data still does not correspond to theoretical assumptions. So the next step is getting rid of all retardation plates and one by one trying out if they are not adding the error or behave differently as they should. One of our plates is replaced for new one and the setup is built again. Additionaly, a 750-nm long-pass filter is mounted on the detector to suppress the influence of the ambient light in the laboratory. The next Figure shows how the enviromental error is lowered and how the measurement changed.

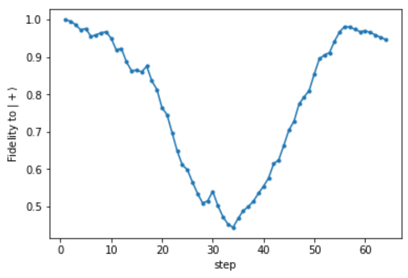


Figure 3.14: Fidelity to diagonal state for $N = 8$

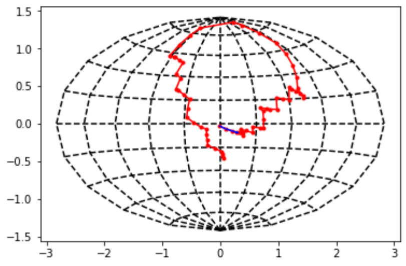


Figure 3.15: The path of the transformations for $N = 8$

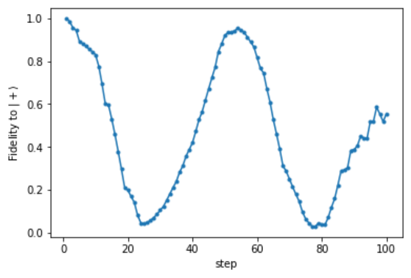


Figure 3.16: Fidelity to diagonal state for $N = 10$

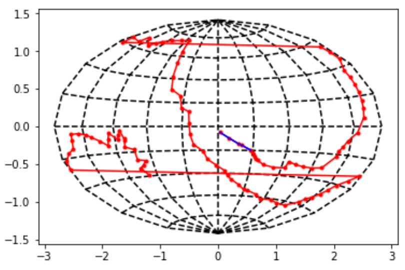


Figure 3.17: The path of the transformations for $N = 10$

These figures show one of the three measurements we performed for both $N = 8$ and $N = 10$. Fidelity is compared with the initial state and therefore we do not want to reach the maximum but the minimum of fidelity. But these measurements still do not agree with the theory and therefore the error has not been compensated yet.

Normaly in all experiments that use retardation plates, these plates have small error that is usually not considered. This error arises from a small inclination of the plate in assembly or from manufacturing imperfections. In our experiment with higher Ns the sequence of measurements gets longer. That means that we are basically using more and more of these plates and their small error must now be considered and compensated. Due to lack of time, colleague Robert Stárek wrote a script in Python that reveals this error and later compensates for it in further measurements. For final measurements his scripts are used and they can be found in Annexes.

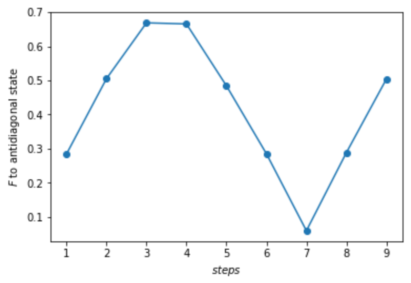


Figure 3.18: Fidelity to antidiagonal state for $N = 3$

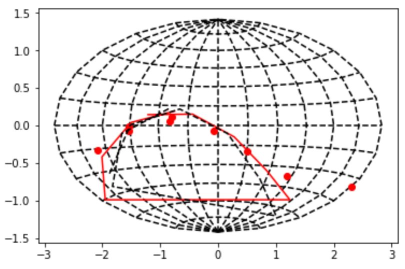


Figure 3.19: The path of the transformations for $N = 3$

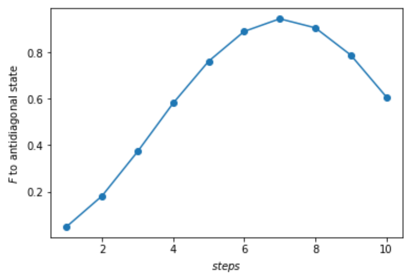


Figure 3.20: Fidelity to antidiagonal state for $N = 5$

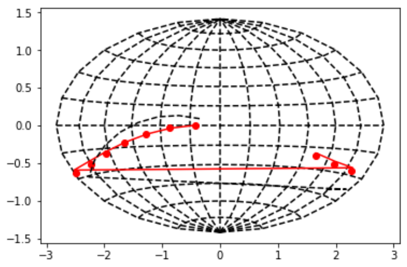


Figure 3.21: The path of the transformations for $N = 5$

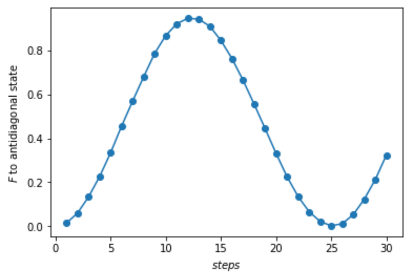


Figure 3.22: Fidelity to antidiagonal state for $N = 6$

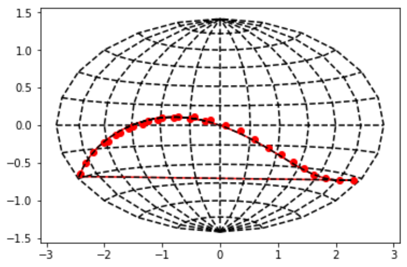


Figure 3.23: The path of the transformations for $N = 6$

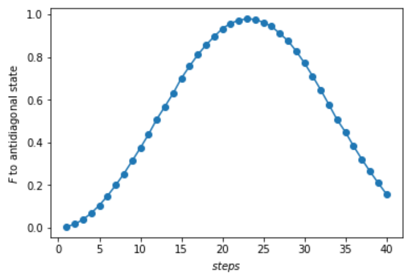


Figure 3.24: Fidelity to antidiagonal state for $N = 9$

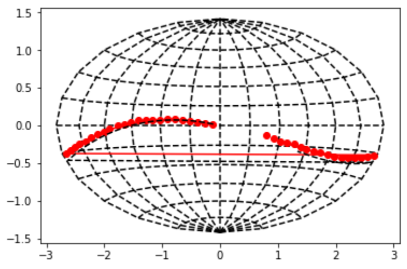


Figure 3.25: The path of the transformations for $N = 9$

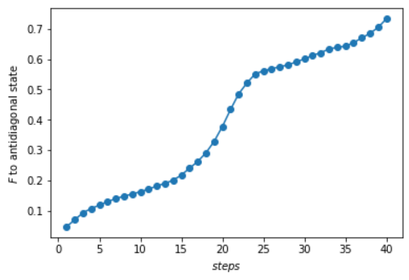


Figure 3.26: Fidelity to antidiagonal state for $N = 10$

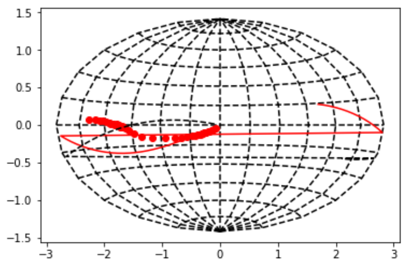


Figure 3.27: The path of the transformations for $N = 10$

In these figures, there are three paths of transformations, where the dashed black line represents the ideal theoretical development according to equation (1.1), the red dots are the measured data and the red line represents calculated evolution from the measured process matrix.

Chapter 4

Conclusions and outlook

The first part of the thesis is based purely on simulation of the whole experiment on the computer. The main result of this part is that we wrote a script that can simulate the experiment for chosen N. We also wrote a script that simulates the relation (1.1) where the small operation is implemented using twelve wave-plates (3.6). We found out that the found formula (3.6) are perfectly matching the theory so we used them to find the correct angles for the next measurements.

Next part was building the setup and mitigating the systematic experimental errors. After bulding the setup we run few tests to see if the setup runs correctly. Later we had to change one of the wave plates. The main problems we encountered were bad angles for preparation and projection, it is impossible to influence the polarization that we bring through the optical fiber because it changes over time and the wrong position of the setup when the plates should rotate all in one direction. Another mistake is that the MaxLik method when reconstructing the density matrix may not be accurate enough and therefore the reconstructed matrix does not completely correspond to the measured data.

In the last part we measured the experiment for few Ns. The the dynamics was measured and we found that small plate errors that are not normally taken into account need to be compensated for their repeated use in one measurement. These errors are mainly manifested for a larger value of N because the measurement sequence is then longer. They manifest themselves because the partial operations we perform are smaller or as large as the plate error, and then these performed partial operations are lost and the setup error prevails. Additionally after the measurement we also need to apply a correction matrix. After we found and compensated for these errors (and optionally applied the correction matrix), we achieved more accurate results that are more in line with the theoretical course of the experiment as shown in Figures 3.18-3.27.

Naturally, there is a space for future improvements. The duration of the quantum state tomography could be reduced with the help of the LCD modules for polarization control [13] and using both outputs from the Glan-Taylor polarizer. Using both outputs could reduce the number of projection settings in tomography and provide a means to norm the photocurrent and eliminate the influence of laser power fluctuations. To avoid the human factor, the waveplates that implement the fixed unitary operation should be mounted in motorized rotation stages. It would also allow automatic adjustment of the waveplates for the compensation of small retardance errors. The search for optimal angles of waveplates should not be a brute-force search. Instead, we would use an optimization algorithm for better performance, as suggested in the later version of the control scripts. To mimic the key idea better, we could also consider using twelve waveplates, instead of just three to implement the rotation operations.

Bibliography

[1] D. E. Deutsch, A. Barenco, and A. Ekert, "Universality in quantum computation," *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, vol. 449, no. 1937, pp. 669–677, 1995.

[2] S. Lloyd and S. L. Braunstein, "Quantum computation over continuous variables," *Phys. Rev. Lett.*, vol. 82, pp. 1784–1787, Feb 1999.

[3] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition.* Cambridge University Press, 2010.

[4] S. Lloyd, "Almost any quantum logic gate is universal," *Phys. Rev. Lett.*, vol. 75, pp. 346–349, Jul 1995.

[5] J. Sakurai and S. Tuan, *Modern Quantum Mechanics.* Advanced book program, Benjamin/Cummings Pub., 1985.

[6] S. Huard, *Polarization of light.* Chichester New York Paris: John Wiley Masson, 1997.

[7] L. Bugayevskiy and J. Snyder, *Map Projections: A Reference Manual.* Taylor & Francis, 1995.

[8] M. Paris, *Quantum state estimation.* Berlin New York: Springer, 2004.

[9] C. Schwemmer, L. Knips, D. Richart, H. Weinfurter, T. Moroder, M. Kleinmann, and O. Gühne, "Systematic errors in current quantum state tomography tools," *Phys. Rev. Lett.*, vol. 114, p. 080403, Feb 2015.

[10] A. Jamiołkowski, "Linear transformations which preserve trace and positive semidefiniteness of operators," *Reports on Mathematical Physics*, vol. 3, no. 4, pp. 275 – 278, 1972.

[11] M.-D. Choi, "Completely positive linear maps on complex matrices," *Linear Algebra and its Applications*, vol. 10, no. 3, pp. 285 – 290, 1975.

[12] B. BARTH, *SMC100CC & SMC100PP - Single-Axis Motion Controller/Driver for DC or Stepper Motor.* Newport, v3.0 ed., dec 2015.

[13] M. Bielak, R. Stárek, V. Krčmarský, M. Mičuda, and M. Ježek, "Accurate polarization preparation and measurement using liquid crystal displays," 2020.

Chapter 5

Annexes

5.1 Setting up HWP and QWP plates

To find the initial "0" angle, you need to have a setup built with all the components except HWP and QWP plates. We place a polarizer in front of the detector, which allows the opposite polarization than we have in the setup. In our case we work with horizontal polarization and therefore the polarizer allows vertical polarization. Then we add a plate, which we either rotate manually or using a program, and try to reach the minimum on the detector. We placed the calibrated waveplate between two crossed linear polarizers in order to find the zero angular position, which corresponds to the minimum intensity in this configuration. If we were looking for the maximum, the measuring device may not be able to detect even small changes around this maximum, which causes inaccurate finding of the initial angle. After finding this minimum and the corresponding angle, leave the plate set at this angle and add another and repeat the procedure.

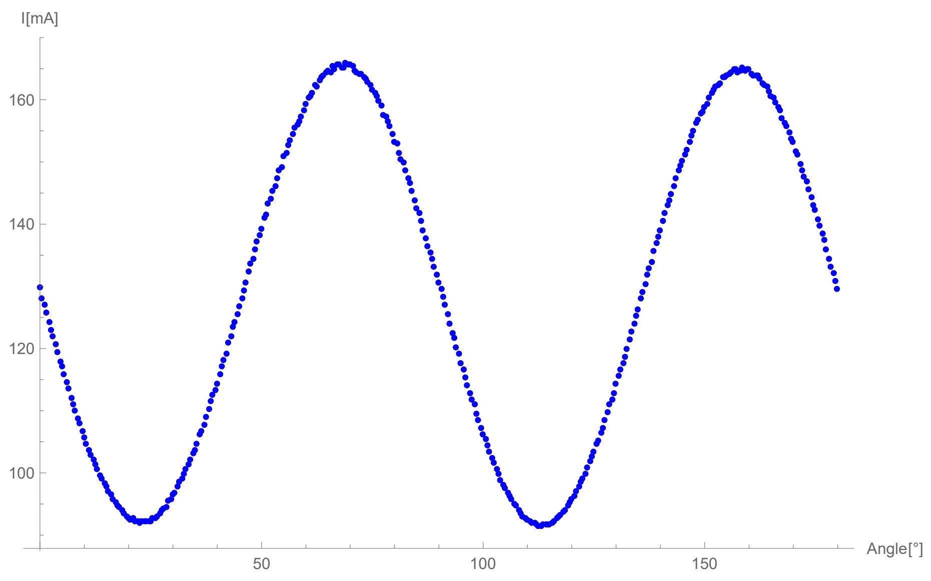


Figure 5.1: Rotation of QWP plate with measured current I on detector

What is important when adding any optical segment to our setup is the reflection. We must try to direct it so that it points as much as possible to the point from where we bring the beam. This ensures the perpendicalar incidence of the beam onto face of the optical component. Another of the problems that can then affect how our setup works is the direction of rotation of the plates, which should be unified. The last of the problems is the fast and slow axis of the plates used. If we rotate the plate by a full 180°, we find two minimums, where one of them will be lower than the other. The problem is that we do not know which of these angles corresponds to the fast and which to the slow axis. This can cause the angles for preparation and projection to be incorrect and will create different states than we want. We will eliminate this error only after the whole setup is built and we can perform tomography for various input states to know if we are preparing and projecting the set states correctly.

5.2 Figures

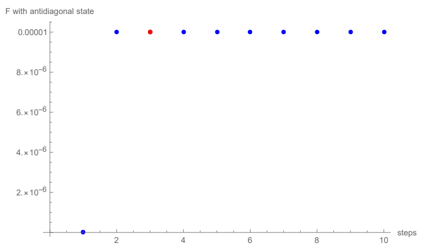


Figure 5.2: Fidelity to antidiagonal state for simulation with $N = 1$, where red dot represents maximum

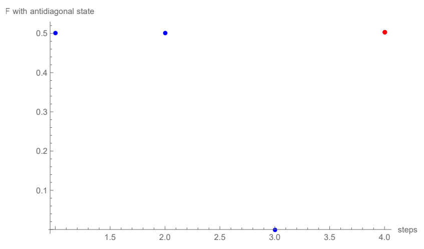


Figure 5.3: Fidelity to antidiagonal state for simulation with $N = 2$, where red dot represents maximum

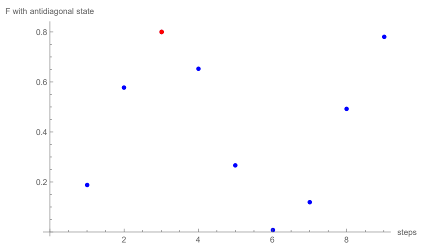


Figure 5.4: Fidelity to antidiagonal state for simulation with $N = 3$, where red dot represents maximum

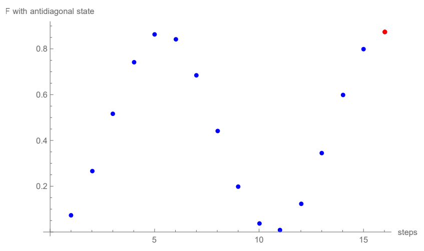


Figure 5.5: Fidelity to antidiagonal state for simulation with $N = 4$, where red dot represents maximum

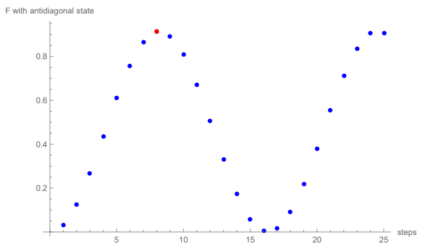


Figure 5.6: Fidelity to antidiagonal state for simulation with $N = 5$, where red dot represents maximum

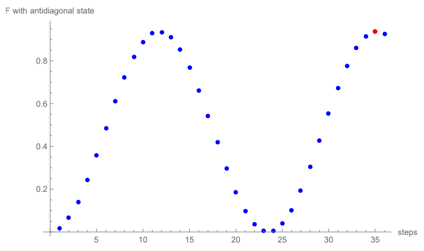


Figure 5.7: Fidelity to antidiagonal state for simulation with $N = 6$, where red dot represents maximum

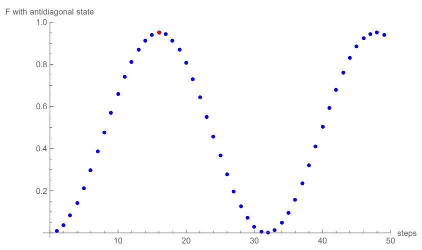


Figure 5.8: Fidelity to antidiagonal state for simulation with $N = 7$, where red dot represents maximum

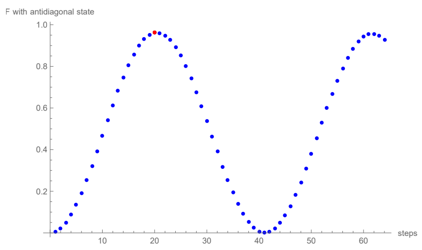


Figure 5.9: Fidelity to antidiagonal state for simulation with $N = 8$, where red dot represents maximum

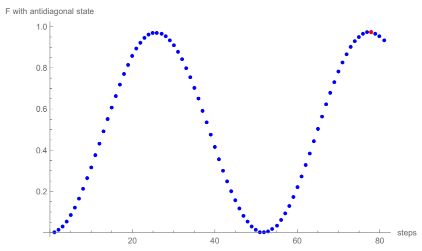


Figure 5.10: Fidelity to antidiagonal state for simulation with $N = 9$, where red dot represents maximum

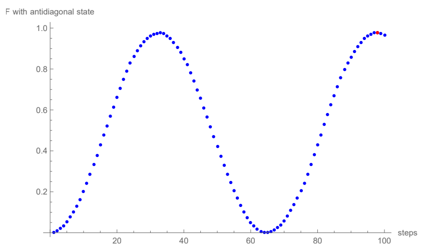
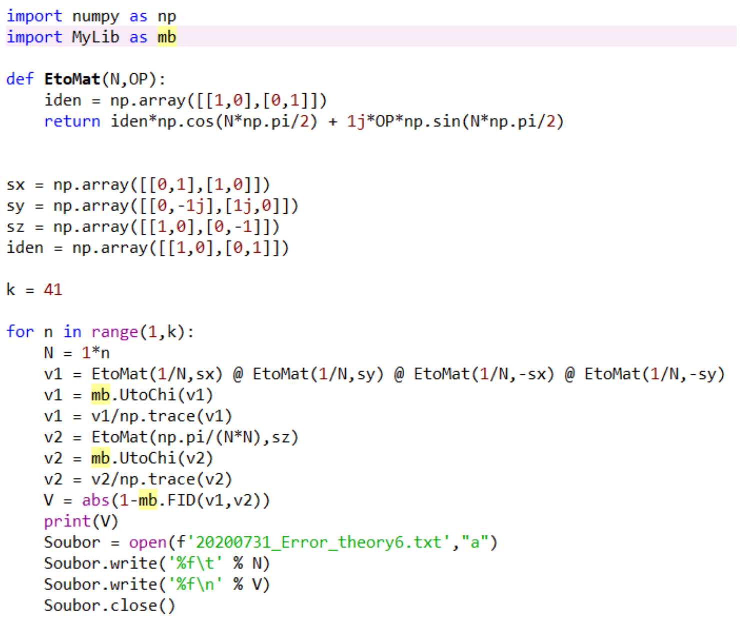
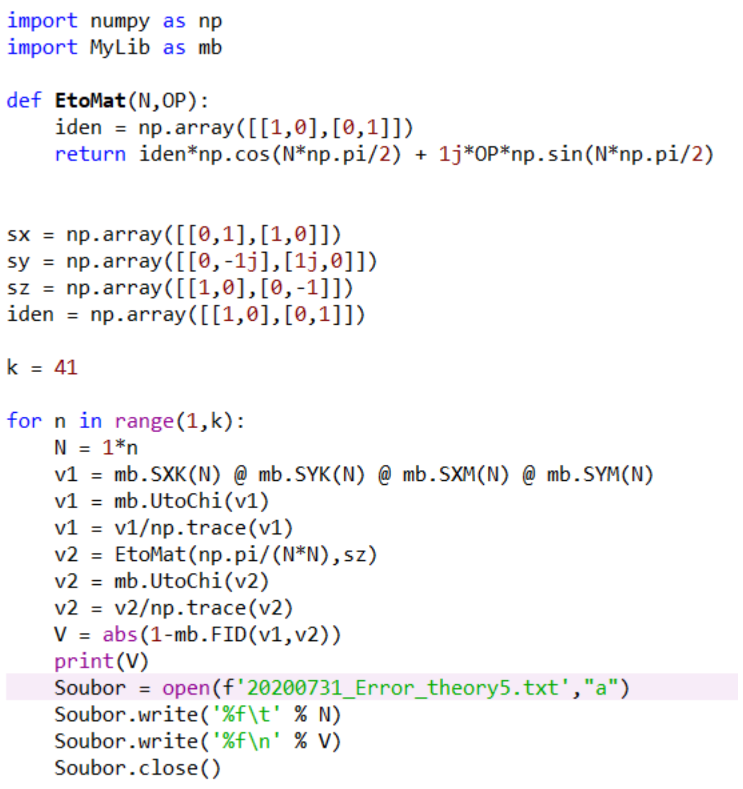


Figure 5.11: Fidelity to antidiagonal state for simulation with $N = 10$, where red dot represents maximum



```
import numpy as np
import MyLib as mb

def EtoMat(N,OP):
    iden = np.array([[1,0],[0,1]])
    return iden*np.cos(N*np.pi/2) + 1j*OP*np.sin(N*np.pi/2)


sx = np.array([[0,1],[1,0]])
sy = np.array([[0,-1j],[1j,0]])
sz = np.array([[1,0],[0,-1]])
iden = np.array([[1,0],[0,1]])

k = 41

for n in range(1,k):
    N = 1*n
    v1 = EtoMat(1/N,sx) @ EtoMat(1/N,sy) @ EtoMat(1/N,-sx) @ EtoMat(1/N,-sy)
    v1 = mb.UtoChi(v1)
    v1 = v1/np.trace(v1)
    v2 = EtoMat(np.pi/(N*N),sz)
    v2 = mb.UtoChi(v2)
    v2 = v2/np.trace(v2)
    V = abs(1-mb.FID(v1,v2))
    print(V)
    Soubor = open(f'20200731_Error_theory6.txt',"a")
    Soubor.write('%f\t' % N)
    Soubor.write('%f\n' % V)
    Soubor.close()
```

Figure 5.12: The script for comparring left and right side of equation (1.1)



```python
import numpy as np
import MyLib as mb

def EtoMat(N,OP):
    iden = np.array([[1,0],[0,1]])
    return iden*np.cos(N*np.pi/2) + 1j*OP*np.sin(N*np.pi/2)


sx = np.array([[0,1],[1,0]])
sy = np.array([[0,-1j],[1j,0]])
sz = np.array([[1,0],[0,-1]])
iden = np.array([[1,0],[0,1]])

k = 41

for n in range(1,k):
    N = 1*n
    v1 = mb.SXK(N) @ mb.SYK(N) @ mb.SXM(N) @ mb.SYM(N)
    v1 = mb.UtoChi(v1)
    v1 = v1/np.trace(v1)
    v2 = EtoMat(np.pi/(N*N),sz)
    v2 = mb.UtoChi(v2)
    v2 = v2/np.trace(v2)
    V = abs(1-mb.FID(v1,v2))
    print(V)
    Soubor = open(f'20200731_Error_theory5.txt',"a")
    Soubor.write('%f\t' % N)
    Soubor.write('%f\n' % V)
    Soubor.close()
```

Figure 5.13: The script for comparring found formula (3.6) and right side of equation (1.1)