Computational Physics III:
Steepest descent, conjugate gradients and SVD

Due on June 08, 2023

Salomon Guinchard

Contents

Problem 1

Steepest descent and conjugate gradient

Say one has the following linear system of equations to solve:

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{1}$$

where $\mathbf{A}$ is square, symmetric and positive definite, and $\mathbf{x}$, $\mathbf{b}$ are column vectors. Then two algorithms can be used to solve the linear system from Eq.(1), namely the steepest descent (SD) and the conjugate gradient (CG) methods. Both methods exploit the fact that solving Eq.(1) for $\mathbf{x}$ is equivalent to extremizing the following quadratic form $\mathbf{f}$:

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

for an arbitrary constant $\mathbf{c}$ since the gradient of $\mathbf{f}$ that we shall denote by $\mathbf{f}'$ recovers Eq.(1):

$$\mathbf{f}'(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}. \tag{3}$$

Thus, one notices that finding the values of $\mathbf{x}$ that extremize $\mathbf{f}$ is equivalent to finding the zeros of the gradient and solving Eq.(1).

(1) Algorithms

The **SD** method starts from an arbitrary $\mathbf{x}^{(0)}$ such that each step of the algorithm gives a point $\mathbf{x}^{(i)}$ closer to the solution $\mathbf{x}$. At each step, the direction of the steepest descent of the quadratic form from Eq.(2) is found and the minimum along this direction is determined, leading to the point $\mathbf{x}^{(i+1)}$. Hence the resulting sequence $\{\mathbf{x}^{(i)}\}$ is decreasing and the convergence rate of the error to the solution $\mathbf{e}^{(i)} := \mathbf{x}^{(i)} - \mathbf{x}$ follows:

$$\frac{|\mathbf{e}^{(i)}|_{\mathbf{A}}}{|\mathbf{e}^{(0)}|_{\mathbf{A}}} \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^i, \tag{4}$$

where $|\mathbf{x}|_{\mathbf{A}} := (\mathbf{x}^T\mathbf{A}\mathbf{x})^{1/2}$ and κ is the so called condition number of $\mathbf{A}$, that is the ratio of the largest singular value of $\mathbf{A}$ to the lowest.

The **CG** method shares basically the same base, the only difference is that the vectors from each step are no longer orthogonal to each other, but $\mathbf{A}$-orthogonal, that is $\mathbf{x}\cdot\mathbf{y} = \mathbf{x}^T\mathbf{A}\mathbf{y}$. This condition turns out to be very efficient, in the sense that if one were to consider an ideal case without numerical imprecision, the algorithme should converge in n steps, with n the size of the matrix. The relative error follows Eq.(5).

Problem 1 [(1) Algorithms] continued on next page...

$$\frac{|\mathbf{e}^{(i)}|_{\mathbf{A}}}{|\mathbf{e}^{(0)}|_{\mathbf{A}}} \leq \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{i} \tag{5}$$

The two algorithms have been implemented in the following scripts and the tests were passed successfully.

Listing 1: Matlab script for the steepest descent method

```
function x = solve_SD(A,b)
% Computes the solution x of Ax=b using the steepest descent method
% INPUT  : A (square symmetric matrix), b (1D vector)
% OUTPUT : x (1D vector)
    N = length(b);
    x = ones(N,1);
    r = b - A*x;
    while norm(r) > 100*eps
        alpha = (r'*r)/(r'*A*r);
        x = x + alpha*r;
        r = b - A*x;
    end
end
```

Listing 2: Matlab script for the conjugate gradient method

```
function x=solve_CG(A,b)
% Computes the solution x of Ax=b using the conjugate method
% INPUT  : A (square symmetric matrix), b (1D vector)
% OUTPUT : x (1D vector)
    N = length(b);
    x = ones(N,1);
    r = b - A*x;
    d = r;
    u = 0;
    while norm(r) > 100*eps
        alpha = (r'*r)/(d'*A*d);
        x = x + alpha*d;
        r_ = r;
        r = r - alpha*A*d;
        beta = (r'*r)/(r_'*r_);
        d = r + beta*d;
        u = u+1;
    end
    disp(u)
end
```

(2) System solving

Using the matrices $\mathbf{A}_i$ and the vectors $\mathbf{b}_i$ $i = 1, 2$ from the file `Matrices.mat`, let us now use the previous algorithms to solve the system from Eq.(1), and plot the relative error to the exact solution at each iteration. Fig.(1) and Fig.(2) show respectively the relative error at each iteration for both the steepest descent and the conjugate gradient methods for two distinct sizes for the matrice $\mathbf{A}$: $N = 5$ and $N = 50$.

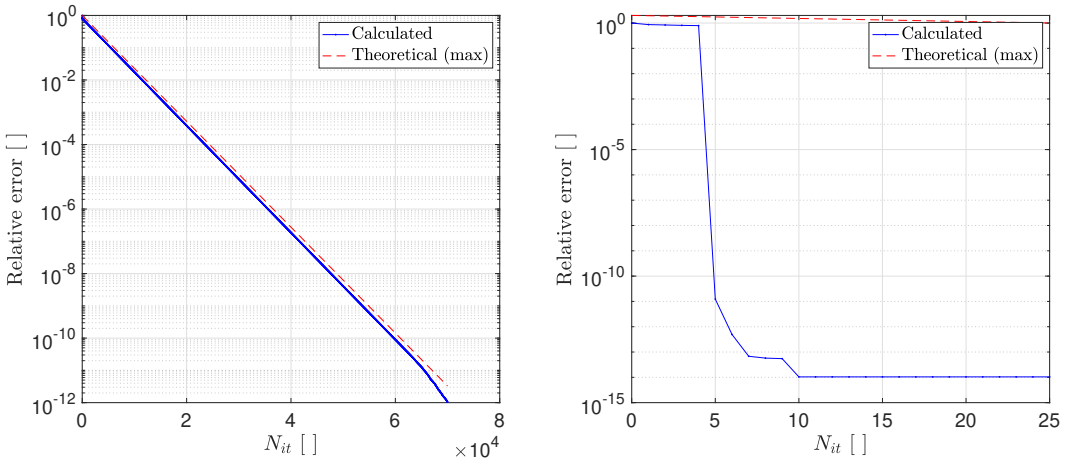


Figure 1: Left: Relative error for $N = 5$ and the steepest descent method, as a function of the number of iterations - Right: Same for the conjugate gradient method

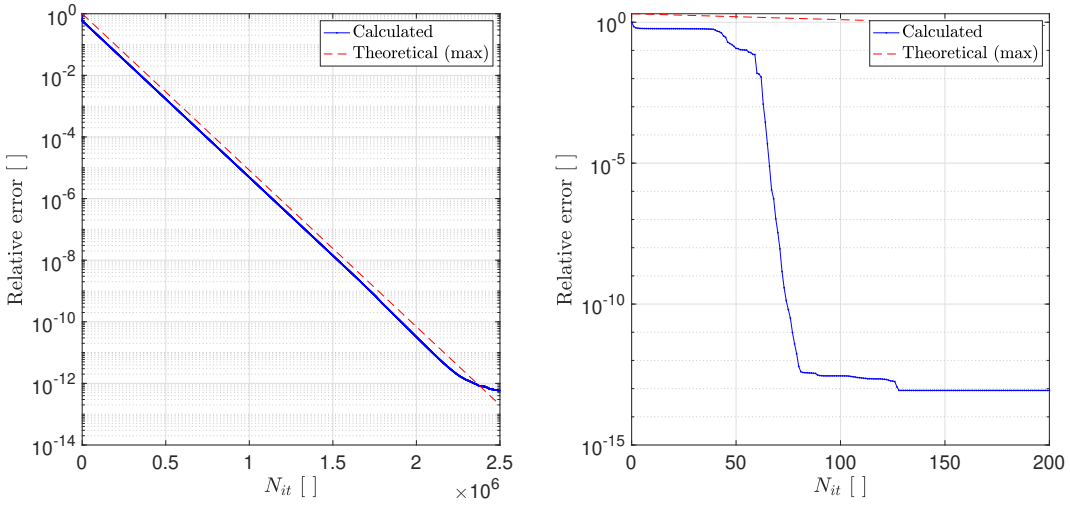


Figure 2: Left: Relative error for $N = 50$ and the steepest descent method as a function of the number of iterations - Right: Same for the conjugate gradient method

(3) Convergence and machine precision tolerance

In the two cases studied previously, with $N = 5$ and $N = 50$, one notes that the conjugate gradient method converges much faster towards the solution than the steepest descent method. Indeed, in the case $N = 5$, the SD method needs about $7 \cdot 10^4$ iterations to converge whereas the CG needs only 10 iterations. In the case $N = 50$, $\sim 2.4 \cdot 10^6$ iterations were needed for the SD to converge against ~ 130 for the CG. In fact, it is possible to obtain a theoretical approximate maximum for the minimum of iterations required to achieve a relative error of the order of the machine precision ϵ. It suffices to invert the relation from Eq.(5) and Eq.(4), yielding the expression $N_{it}^{\epsilon} = \ln(\epsilon)/\ln((\kappa-1)/(\kappa+1))$ for the SD method, where κ is the condition number of the matrix. Hence, in the case of $N = 5$ (SD), one gets that $N_{it}^{\epsilon} = 95421 \sim 9 \cdot 10^4$ and for $N = 50$, $N_{it}^{\epsilon} = 3081446 \sim 3 \cdot 10^6$ which are compatible results with those observed on Fig.(1) and Fig.(2).

Problem 2

Discretization of Poisson Equation

Let us define the following system, consisting in a square grid of size $L \times L$ with $L = 45$. Let a charge $Q = 1\text{C}$ at the center of the grid. Two electrodes of length $L/3$, each kept at the potential of $V_0 = 1\text{V}$ are disposed (and centered) on the left and right sides. The remaining boundary is insulated, that is no electric field $\mathbf{E} = \nabla V$ perpendicular to the boundary is present elsewhere. Recall that if the medium is devoid of magnetic field or electrical current, the electric potential $V(x, y)$ satisfies Poisson's equation:

$$\Delta V(x, y) = -\frac{\rho(x, y)}{\epsilon} \tag{6}$$

The electric potential resulting from that test charge and the electrodes, setting $\epsilon = 1$, will now be determined using the SD and CG methods derived previously.

(1) Discretization

To study the system and solve Poisson's equation, one has to discretize the problem so that it can be recast in the form of a system of linear equations $\mathbf{A}A\mathbf{x} = \mathbf{b}$. To do so, a grid of size $N \times N$ is considered, with $N = 45$. Therefore, the discretized system is composed of N^2 points (x_i, y_j). Let us now apply the following change of coordinates $(x_i, y_j) \to r_k$, $k \in \{1, N^2\}$ so the discretisation of the Laplacian operator is easier. Thus the matrix $\mathbf{A}$ now becomes a matrix of size $N^2 \times N^2$, and $\mathbf{x}$ takes the following form $\mathbf{x} = (V_1, ..., V_{N^2})^T$. Moreoever, the Laplacian operator has been discretized using second order finite differences:

$$\nabla^2 V(r_k) \approx \frac{V_{k+1} + V_{k-1} + V_{k+N} + V_{k-N} - 4V_k}{h^2} \tag{7}$$

where $V_k \equiv V(r_k)$ and h is the mesh parameter, that is the distance between to neighboring points on the grid. Note that Eq.(7) only holds for the interior of the system, and not on the boundary. Indeed, take for example the left edge of the system, without considering the corners, which have to be treated separately. In that case, the discretized Laplacian becomes:

$$\nabla^2 V(r_k) \approx \frac{V_{k+1} + V_{k+N} + V_{k-N} - 4V_k}{h^2} \tag{8}$$

The same reasoning holds for the other edges of the system, as well as for the 4 corners. Now, let us focus on the implementation of the boundary conditions (BC). First, let us recall that the matrix $\mathbf{A}$ is assumed to be symmetric and positive definite, since the SD and CG algorithms will be used to solve the system. Therefore, implementing the BC in the matrix $\mathbf{A}$ would break the symmetry required by the SD and CG methods. This constraint can however be respected by transferring the BC into the right-hand side of the Poisson equation, i.e. into the charge density term ρ (see Eq.(6). Since $\epsilon = 1$ and the charge is 1C, the potential at the center of the grid is fixed at 1V. This whole approach was implemented in the file `discretized.m`.

The electric field $\mathbf{E}$, defined by the relation $\mathbf{E}(x,y) = -\nabla V(x,y)$ can also be numerically implemented and visualized as a vector field. In this case, the gradient operator has been implemented by mean of finite differences of order 1:

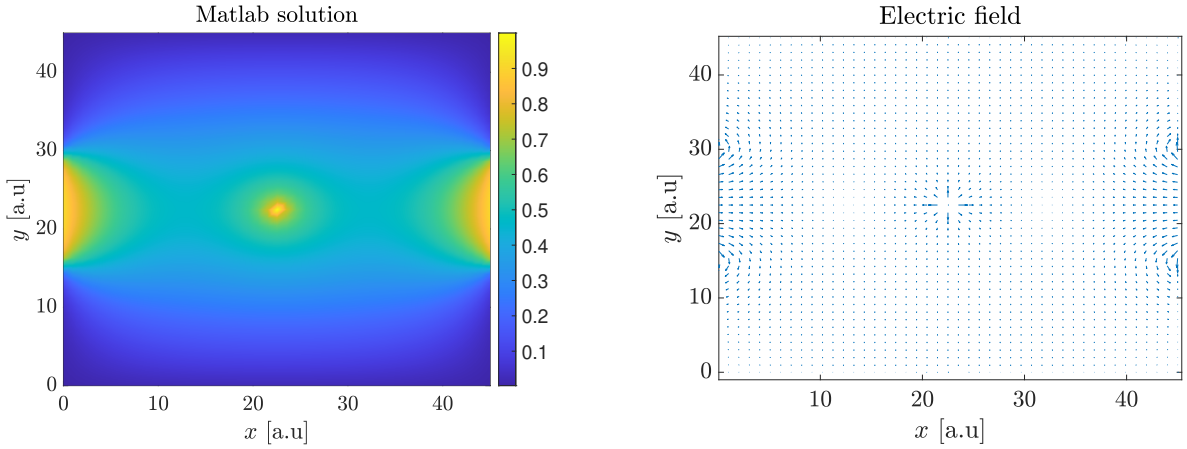
$$E^x_{i,j} = -\frac{V_{i,j+1} - V_{i,j-1}}{h}, \qquad E^y_{i,j} = -\frac{V_{i+1,j} - V_{i-1,j}}{h} \tag{9}$$

Note that here again, one has to be careful when implementing the BC. Indeed, insulators and conductors are present on the edges of the system. Therefore, the condition that $\mathbf{E}^{\perp} = 0$ along the insulated regions has to be implemented. For conducting regions, $\mathbf{E}$ can be implemented using so-called "forward" and "backward" finite differences (see `Efield.m`).

$$\text{Forward}: \quad E^x_{i,j} = -\frac{V_{i,j} - V_{i,j-1}}{2h}, \qquad \text{Backward}: \quad E^x_{i,j} = -\frac{V_{i,j+1} - V_{i,j}}{2h} \tag{10}$$

(2) Numerical solution

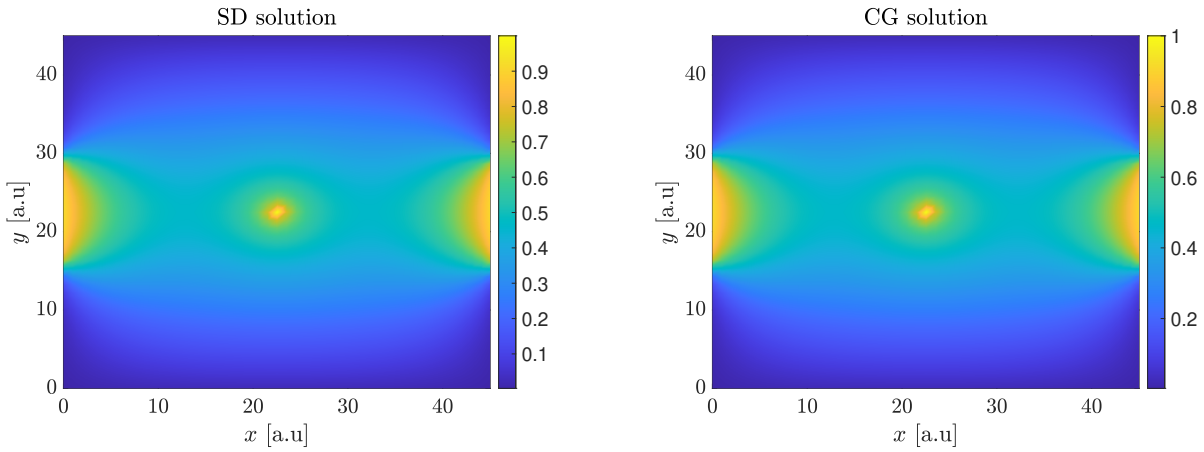
Let us present the numerical solution now that $\mathbf{A}$ and the vector $\mathbf{b}$ have been defined. The solutions obtained through the SD and CG methods are compared with the solution provided by Matlab according to the Matlab command `x_M=A\b`. The Matlab solution is shown in Fig.(3a), the $\mathbf{E}$-field is shown in Fig.(3b). The solutions obtained with the SD and CG methods are shown in Fig.(4a) and Fig.(4b), respectively.



(a) (b)

Figure 3: Left: solution for the electric potential given by Matlab - Right: Quiver plot of the electric field for that same configuration

(3) Electric potential



(a) (b)

Figure 4: Left: Electric potential obtained by the SD algorithm - Right: Same for the CG algorithm

It was also possible to determine N_{it} the number of iterations required to get to those solutions. In the case of the SD method, 7′890 iterations were required to converge to a solution whose error w.r.t Matlab's solution is $3.5 \cdot 10^{-11}$. As of the CG method, only 239 iterations were required. The error w.r.t Matlab's solution is $1.7 \cdot 10^{-15}$, which is not far from machine precision.

Note that the solutions obtained are physically consistent. Indeed, the obtained electric potential is 1V at the centre of the grid where a charge of 1C has been placed. Moreover, the electric field moves from areas of high electric potential (conductors) to areas of low electric potential (insulators), which is consistent with the laws of electromagnetism.

(4) Jacobi preconditioning

The main idea of preconditioning is to solve the system $\mathbf{Ax} = \mathbf{b}$ indirectly, i.e. by solving $\mathbf{M}^{-1}\mathbf{Ax} = \mathbf{M}^{-1}\mathbf{b}$. Different preconditioning can be implemented, such as the Jacobi preconditioning for example, where one defines the matrix $\mathbf{M}$ as diagonal, formed by the diagonal elements of A. As expected, the Jacobi preconditioning accelerated the convergence of the SD and CG methods. Indeed, the number of iterations required for the convergence were now 7′430 iterations for the SD and 220 for the CG methods.

Problem 3

Nonlinear conjugate gradient method

Consider an ensemble of N identical atoms interacting with each other by means of a pair potential

$$E(\mathbf{x}_1, \dots, \mathbf{x}_N) = \sum_{i \neq j} U_{LJ}(|\mathbf{x}_i - \mathbf{x}_j|) \tag{11}$$

where $\{\mathbf{x}_i\}$ are the coordinates of atoms and $U_{LJ}(\mathbf{x})$ is the Lennard-Jones potential which describes weak van der Waals interaction

$$U_{LJ}(x) = 4\epsilon \left[\left(\frac{\sigma}{x}\right)^{12} - \left(\frac{\sigma}{x}\right)^{6} \right] \tag{12}$$

The minima on the potential energy surface described by $E(\mathbf{x}_1, \dots, \mathbf{x}_N)$ correspond to equilibrium configurations of Lennard-Jones potential. This potential is implemented in the file `LennnardJones.m`.

(1) Algorithm

The methods previously used (SD and CG) were used to find the minima of a quadratic form in order to solve the associated linear problem. However, some physics situations require to solve more complicated problems, in the sense that they wouldn't necessarily be linear. That is a reason to introduce the non-linear conjugate gradient (NLCG) method.

This algorithm, implemented in the `non_linCG.m` file, is based on the CG algorithm, but different regarding what follows. A recursive formula in the residual calculation cannot be used. The residual is then constantly fixed at the opposite of the gradient, that is $r_i = -f'(x_i)$. Moreover, the β coefficient is calculated thanks to Fletcher-Reeves formula, i.e. $\beta_{i+1} = r_{i+1}^t r_{i+1}/(r_i^t r_i)$. Finally, finding α the step size is somehow more complicated. To overcome this difficulty, a Newton-Raphson method is used. The main idea is to minimize, using a Taylor expansion of order 2, the expression $f(\mathbf{x} + \alpha\mathbf{d})$ along the vector $\mathbf{d}$, as it has been shown in the lecture notes. In that problem, the functional to minimize, f, is the energy E as defined in Eq.(11).

The Newton-Raphson method writes

$$E(\mathbf{x} + \alpha\mathbf{d}) \sim E(\mathbf{x}) + \alpha(E'(\mathbf{x}))^t + \frac{\alpha^2}{2}\mathbf{d}^t E''(\mathbf{x})\mathbf{d} \tag{13}$$

where $E''(\mathbf{x})$ is the Hessian matrix, containing the second order derivatives of E with respect to the coordinates. Note that it only needs to be evaluated in the direction $\mathbf{d}$ (see `Hessian.m` file). Finally, when the second derivative is negative, the step is taken in the direction of the maximum rather than the minimum. Thus the idea is to take a small step of $-\epsilon E'(\mathbf{x})$ down the gradient.

Test of the algorithm

In order to test NLCG algorithm, the cases of $N \in \{2, 3\}$ have been implemented. For these two configurations, the equilibrium positions were determined thanks to the NLCG method (see Fig.(5a) and Fig.(5b)). The total energy E of the equilibrium configuration was also determined. These are given by $E_2 = -1$ and $E_3 = -3$.

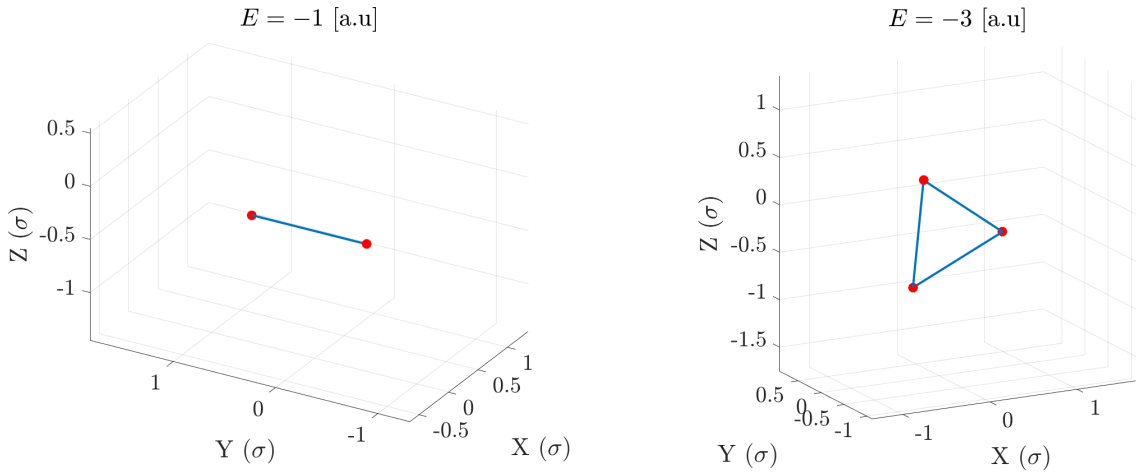


Figure 5: Left: final configuration minimizing the energy functional in the case of 2 atoms - Right: same for 3 atoms

Those obtained equilibrium configuration seem to agree with the Lennard-Jones potential. Indeed, the term to the 6th power in Eq.(12) accounts for the attraction between atoms and dominates at long distances, while the term to the 12th power is an empirical term that accounts for Pauli's exclusion principle.

(2) $N = 4$

Consider now the case of a molecule of 4 atoms. Two distinct equilibrium configurations are possible: the square planar geometry and the tetrahedral geometry (see 6a and 6b, respectively).

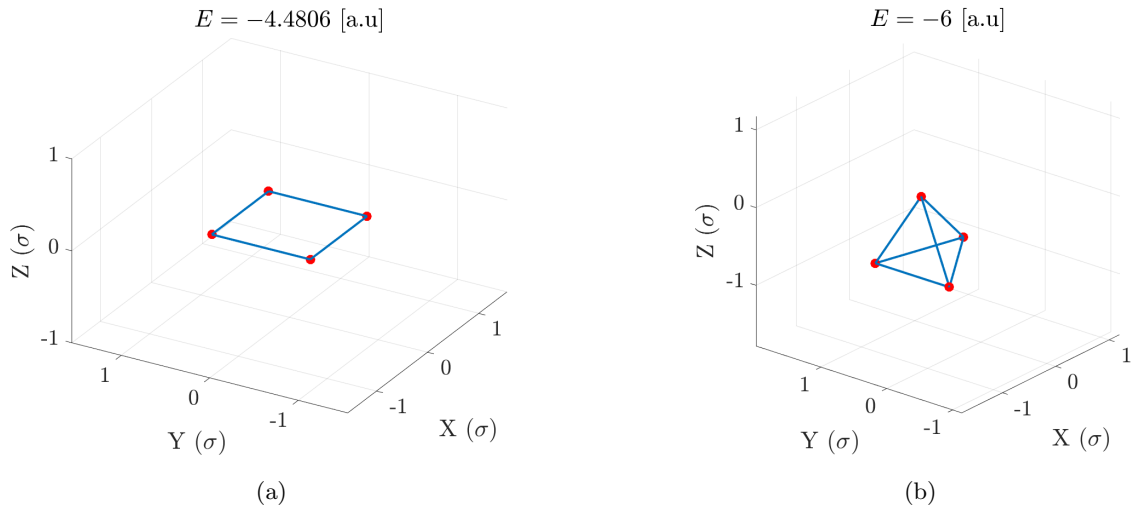


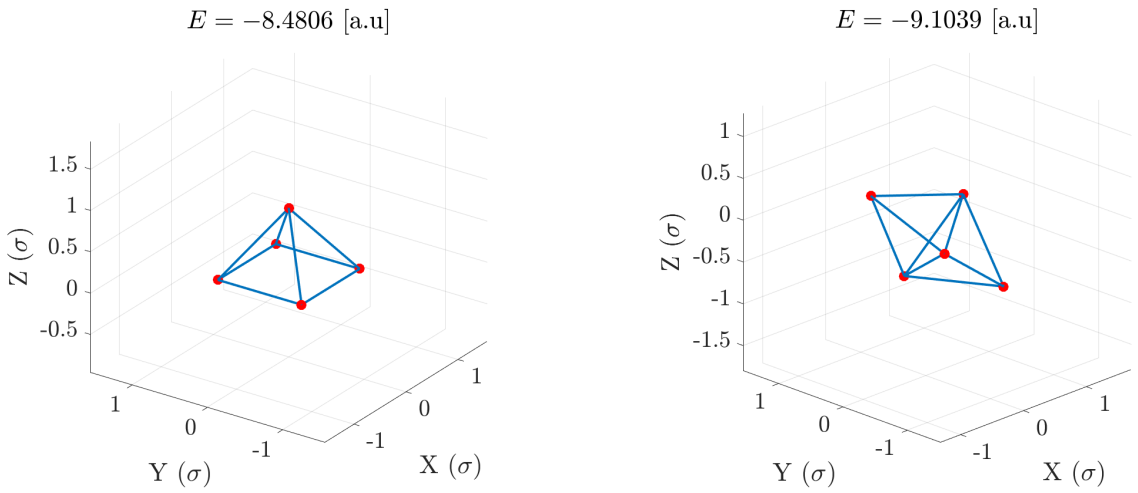
Figure 6: Equilibrium configurations in the case of 4 atoms.

In order to obtain the tetrahedral configuration, the atoms were initially disposed randomly. The total energy of the equilibrium configuration is negative and is $E_{4,t} = -6$. Since the energy is negative, this state is a stable equilibrium bound state. To recover the flat square equilibrium configuration, the initial positions must not be set randomly. In fact, the tetrahedral configuration is more stable than the planar square configuration, which is then favorable in the case of random initial positions. However, any rectangular or square initial configuration will converge to a plane square equilibrium configuration, whose total energy

is $E_{4,s} = -4.4806$. Thus, the square equilibrium is also a stable bound state, but less stable than the tetrahedral configuration ($E_4^t < E_4^s$).

(3) $N = 5$

The same analysis is now carried out on a system made up of 5 atoms. Clusters of 5 atoms can arrange either in trigonal bi-pyramid (see Fig.(7b)) or a square pyramid geometry (see Fig.(7a)).



(a) (b)

Figure 7: Left: quare pyramid equilibrium configuration ($N = 5$ atoms) - Right: trigonal equilibrium

To obtain the previous equilibrium configurations, the same approach as for $N = 4$ was used. A set of random positions is used to obtain a trigonal bi-pyramidal equilibrium. The total energy of this equilibrium configuration is $E_{5,t} = -9.1039$, characteristic of a stable bound state. To get the square pyramid equilibrium configuration, any configuration whose pyramidal base is flat and whose last atom is positioned along the axis perpendicular to the plane formed by the first 4 atoms converges towards a square pyramidal equilibrium configuration. A scan on various positions following that configuration has been made. In that case, the total energy of the system is given by $E_{5,s} = -8.4806$. Thus the square pyramid configuration also corresponds to a stable bound state. Note that this configuration can be considered less stable as the trigonal bi-pyramid configuration since $E_{5,t} < E_{5,s}$.

(4) $N = 6$

Increasing the number of atoms lead to more diverse equilibrium configurations. The first one is given by the trigonal prismatic geometry. This configuration is obtained by placing three atoms on the vertices of a triangle.For convergence speed purposes, the triangle was chosen to be equilateral. In addition, one atom is placed on a straight line perpendicular to the plane formed by the first 3 atoms. The last two atoms are placed on the opposite side of the plane formed by the first 3 atoms. The resulting equilibrium's energy is of $E_{6,t} = -12.3029$. The equilibrium configuration is pictured in Fig.(8a)

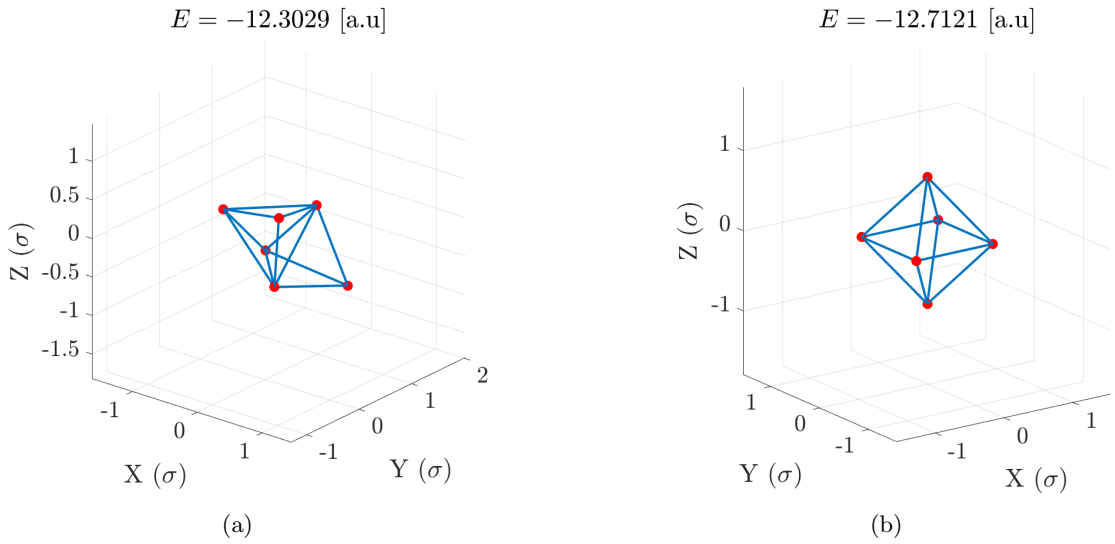


Figure 8: Left: trigonal equilibrium configuration ($N = 6$ atoms) - Right: bipyramid equilibrium

The second one considered here is the octahedral geometry. This configuration can be obtained by imposing a plane base of 4 atoms, square or rectangular, and placing the two remaining atoms on a straight line perpendicular to that plane, on either side of the latter (equally spaced or not). Fig.(8b) shows this equilibrium configuration for a total energy of $E_{6,o} = -12.7121$.

Problem 4

SVD: Over-defined system of linear equations

(1) SVD - method

The Singular Value Decomposition (SVD) consists in the decomposition of a real or complex matrix $\mathbf{A}$ of size $m \times n$ ($m \geq n$) in the following product $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\star$, where $\mathbf{U}$ is a $m \times m$ unitary matrix of left-singular vectors, $\mathbf{V}$ is a $n \times n$ unitary matrix of right-singular vectors and $\mathbf{\Sigma}$ is a $m \times n$ diagonal matrix of singular values. The SVD can be very useful to solve over-defined system of linear equations such as the one from Eq.(14).

$$\begin{pmatrix} 3 & 2 \\ 4 & 5 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 4 \\ 1 \end{pmatrix} \iff \mathbf{Ax} = \mathbf{b}. \tag{14}$$

Since the system is over-defined (in our case 3 equations for 2 unknowns), the idea is to minimize the residual r, defined as $r(\mathbf{x}) = ||\mathbf{Ax} - \mathbf{b}||$. If A is a full-rank matrix, then the solution of the system of linear equation can be written as $\mathbf{x} = \mathbf{A}^+\mathbf{b}$, where $\mathbf{A}^+$ denotes the Penrose inverse of the matrix $\mathbf{A}$. The Penrose inverse of $\mathbf{A}$ can be obtained running the Matlab command `pinv(A)`. In that case, it yields the following result for $\mathbf{x}^* := \mathbf{x}$ such that $r(\mathbf{x})$ is min: $\mathbf{x}^* = (x_1^*, x_2^*) = (-0.4118, 1.1373)$.

(2) SVD - example

We verify easily that the result given in **(1)** is correct by computing the residual in the plane (x_1, x_2) around (x_1^*, x_2^*). The solution obtained lies indeed at the minimum of the paraboloid formed by the contours of the residual, as depicted in Fig.(9)

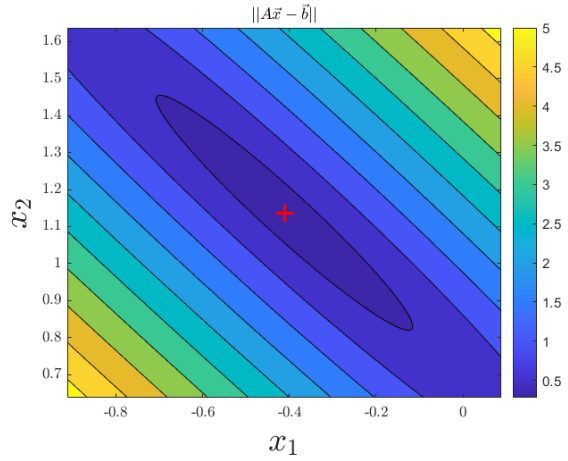


Figure 9: Contours of the residual $r(\mathbf{x})$ in the vicinity of $\mathbf{x}^*$. The red + marks the position of $\mathbf{x}^*$

Problem 5

SVD: Quantum state tomography

The density matrix formalism

In order to describe quantum systems, it is of crucial importance to be able to express statistical distribution of quantum states. The density matrix formalism provides a tool for such a study. Let us consider a statistical mix of N states $|\psi_n\rangle$, each with probability p_n such that $\sum_n p_n = 1$. The density matrix describing this mixture is defined as:

$$\hat{\rho} = \sum_n p_n |\psi_n\rangle \langle\psi_n| \tag{15}$$

(1) Diagonalization

The first part of this exercise deals with the diagonalization of the following density matrices:

$$\hat{\rho_1} = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \qquad \hat{\rho_2} = \frac{1}{2}\begin{pmatrix} -3 & -1 \\ -1 & 1 \end{pmatrix} \tag{16}$$

Let us denote the resulting diagonal matrices by D_1 and D_2, respectively.

$$D_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \qquad D_2 = \begin{pmatrix} \frac{2+\sqrt{2}}{2} & 0 \\ 0 & \frac{2-\sqrt{2}}{2} \end{pmatrix} \tag{17}$$

Recall that in quantum mechanics, the state of a system is described by an element $|\psi\rangle$ from the Hilbert space associated with the system. Such a state can be described by the density operator:

(2) Density matrix properties

$$\hat{\rho} = |\psi\rangle \langle\psi| \tag{18}$$

If a density operator is of this form, it satisfies $\hat{\rho}^2 = \hat{\rho}$. Conversely, if $\hat{\rho}^2 = \hat{\rho}$, then there exists $|\psi\rangle$ such that $\hat{\rho} = |\psi\rangle \langle\psi|$. In that case, ρ is said to describe a *a pure state*. This result can be proven as follows. If $\hat{\rho}^2 = \hat{\rho}$, then in the basis that diagonalizes $\hat{\rho}$, it comes that $\lambda_i = 0$ or 1. But $\sum_i \lambda_i = 1$. Therefore, there exists n such that $\lambda_n = 1$ and $\lambda_j = 0$ for $j \neq n$. This implies that $\hat{\rho} = |\psi_n\rangle \langle\psi_n|$. Hence, the $|\psi_n\rangle$ state is the eigenvector of the density matrix associated with the non-zero λ_n eigenvalue.

It can be verified that $\hat{\rho}_1^2 = \hat{\rho}_1$. As a result, there is a state $|\psi\rangle$ such that $\hat{\rho} = |\psi\rangle\langle\psi|$. This state is given by $|\psi\rangle = \frac{1}{\sqrt{2}}(|+\rangle + |-\rangle)$, where $|+\rangle$ and $|-\rangle$ are the eigen-kets of $\hat{\sigma}_z$.

In addition, the density matrix has the following properties: $\hat{\rho}$ is Hermitian, with trace equal to 1. Therefore, not every 2×2 matrix can be a valid density matrix. However, considering the generic decomposition of a 2×2 matrix following:

$$\hat{\rho} = \sum_{n=0}^{3} \rho_n \hat{\sigma}_n \tag{19}$$

where $\rho_n \in \mathbb{C}$, $\hat{\sigma}_0 \equiv \hat{1}$ and $\hat{\sigma}_n$, $n \neq 0$, are the Pauli matrices. Thus, constraints on ρ_n can be established to ensure that the properties of a density matrix are satisfied. The matrix from Eq.(19) can be calculated explicitly, giving

$$\hat{\rho} = \begin{pmatrix} \rho_0 + \rho_3 & \rho_1 - i\rho_2 \\ \rho_1 + i\rho_2 & \rho_0 - \rho_3 \end{pmatrix} \quad \Longrightarrow \quad \begin{cases} \mathrm{Tr}(\hat{\rho}) = 1 & \Leftrightarrow \quad \rho_0 = 1/2 \\ \hat{\rho}^\dagger = \hat{\rho} & \Leftrightarrow \quad \rho_1, \rho_2, \rho_3 \in \mathbb{R} \end{cases} \tag{20}$$

Simple Quantum state tomography

(1) Density matrix reconstruction

The goal of the quantum tomography is to reconstruct the density matrix $\hat{\rho}$. Generally, a single observable does not provide enough information to fully reconstruct the density matrix. To generalize this approach, let us consider $M/2$ observables, yielding M projection operators defined as follows:

$$\hat{E}_m \equiv |\psi_m\rangle\langle\psi_m| \tag{21}$$

In a similar way, let p_m be the probability associated with $\hat{E}_m$. Thus, the following system of equations $p_m = \mathrm{Tr}(\hat{\rho}\hat{E}_m)$, $m = 1, ..., M$, has to be solved. With the decomposition from Eq.(19), this system of equations can be cast into a matrix equation according to $\mathbf{p} - \mathbf{M}_0 = \mathbf{M}\rho$, where

$$\mathbf{p} = (p_1, \ldots, p_M)^T, \quad \mathbf{M}_0 = (1/2, ..., 1/2)^t, \quad \rho = (\rho_1, \rho_2, \rho_3)^T, \quad (\mathbf{M})_{m,i} = \mathrm{Tr}(\hat{\sigma}_i \hat{E}_m),\ i \in \{1, 2, 3\} \tag{22}$$

The matrix $\mathbf{M}$ is the so-called *measurement matrix*. Using the Penrose inverse (`pinv` command in Matlab) and the basis formed by the $|n\pm\rangle$ vectors, i.e. the eigenvectors of the $\hat{\sigma}_n$ operator, it is possible to reconstruct the density matrix for a given probability vector $\mathbf{p}$. These vectors are given by the files `p1.csv` and `p2.csv`. Note that by convention, $|\psi_{1,2}\rangle = |x\pm\rangle$, $|\psi_{3,4}\rangle = |y\pm\rangle$ and $|\psi_{5,6}\rangle = |z\pm\rangle$. The matrix $\mathbf{M}$ was constructed using the matlab function from `measurement.m`. Solving the equation $\mathbf{p} - \mathbf{M}_0 = \mathbf{M}\rho$ using

the Penrose inverse, which is possible since the rank of M is 3, gives the following ρs vectors, associated with the probability vectors $\mathbf{p}_1$ and $\mathbf{p}_2$, respectively

$$\vec{\rho_1} = \begin{pmatrix} 0.3536 \\ 0 \\ 0.3536 \end{pmatrix}, \qquad \vec{\rho_2} = \begin{pmatrix} 0.2439 \\ 0.3 \\ 0.2707 \end{pmatrix} \tag{23}$$

Finally, combining Eq.(19) and Eq.(23), it follows that the reconstructed density matrices are:

$$\hat{\rho}_1 = \begin{pmatrix} 0.8536 & 0.3536 \\ 0.3536 & 0.1464 \end{pmatrix}, \qquad \hat{\rho}_2 = \begin{pmatrix} 0.7707 & 0.2439 - 0.3i \\ 0.2439 + 0.3i & 0.2293 \end{pmatrix} \tag{24}$$

One can be observe that for both $\hat{\rho}_1$ and $\hat{\rho}_2$, $\mathrm{Tr}(\hat{\rho}_i) = 1$, $i \in \{1, 2\}$. Moreover, $\hat{\rho}_i = \hat{\rho}_i^\dagger$ is satisfied. In addition, it can also be verified that $\hat{\rho}_1^2 = \hat{\rho}_1$, but $\hat{\rho}_2^2 \neq \hat{\rho}_2$. Thus, there is a state $|\phi\rangle$, such that $\hat{\rho}_1 = |\phi\rangle \langle\phi|$. This state is given by the eigenvector of $\hat{\rho}_1$ associated to its non-zero eigenvalue:

$$|\phi\rangle = \begin{pmatrix} -0.9239 \\ -0.3827 \end{pmatrix} \tag{25}$$

(2) Reconstruction with less projection operators

The same analysis was carried out, but this time only with the first 4 projection operators, i.e. $|\psi_{1,2}\rangle = |x\pm\rangle$ and $|\psi_{3,4}\rangle = |y\pm\rangle$. In this case, the singular values of the measurement matrix are given by $\sigma_1 = \sqrt{2}$, $\sigma_2 = \sqrt{2}$ and $\sigma_3 = 0$. As a result, the M measurement matrix is not full-rank. In other words, the matrix M is singular, and the Penrose inverse cannot be used to solve the equation $\vec{p} - \vec{M}_0 = M\vec{\rho}$.

Quantum state tomography with experimental constraints

(1) Reconstruct the density matrix?

Experimentally, it can happen that the measurement can only be performed along a direction specified by two angles θ and ϕ, yielding:

$$|\psi(\theta, \phi)\rangle \equiv \begin{pmatrix} \cos\theta/2 \\ e^{i\phi} \sin\theta/2 \end{pmatrix}. \tag{26}$$

The projection operator is, as before, defined by $\hat{E} = |\psi(\theta, \phi)\rangle \langle\psi(\theta, \phi)|$. However, a single observable is not enough to reconstruct the density matrix. Thus, let us consider a single fermion subject to an external magnetic field along the $\hat{z}$ direction. Assuming that the Hamiltonian $\hat{H}$ is given by $\hat{H} = \hbar\Omega\hat{\sigma_z}$, with Ω the

angular frequency of the magnetic field, it can be shown that

$$e^{i\hat{H}t/\hbar}\left|\psi(\theta,\phi)\right\rangle = \left|\psi(\theta,\phi-\Omega t)\right\rangle \tag{27}$$

In other words, measuring at different times is equivalent to rotating the axis. This rotation allows to gain a lot more information about the system, but is not guaranteed to be sufficient to fully reconstruct the density matrix.

The first step is to determine by mean of SVD, whether or not reconstruction of the density matrix is possible. To this end, various properties of the SVD will be exploited, in particular the fact that the number of non-zero singular values represents the rank of the considered matrix. This information will enable to determine if the matrix is full rank or not.

To that end, let us consider the following set of projection operators : $\left|\psi(\theta,\phi_m)\right\rangle\left\langle\psi(\theta,\phi_m)\right|$, where $\phi_m = 2\pi k/M$, $k = 0, ..., M-1$. This is equivalent to a measurement along the $(\theta, \phi = 0)$ axis, for M different times, uniformly distributed between $t = 0$ and $t = 2\pi/\Omega$. The SVD was carried out on the measurement matrix (see Eq.(22)) for $\theta \in \{0, \pi/2, \pi/3\}$ and $M = 100$. The singular values obtained have been arranged in a vector $\vec{v}_{\theta_i}$ to make it easier to read. The index θ_i corresponds to the angle θ considered.

$$\vec{v}_0 = \begin{pmatrix} 10 \\ 0 \\ 0 \end{pmatrix}, \quad \vec{v}_{\pi/2} = \begin{pmatrix} 7.1063 \\ 7.0356 \\ 2.08 \cdot 10^{-15} \end{pmatrix}, \quad \vec{v}_{\pi/3} = \begin{pmatrix} 6.1554 \\ 6.0930 \\ 4.99 \end{pmatrix} \tag{28}$$

It can therefore be concluded that the density matrix can only be reconstructed here when $\theta = \pi/3$. Indeed, in the case where $\theta = 0$, it is clear that there are zero singular values. In the case where $\theta = \pi/2$, the singular value of the order of 10^{-15} is of the order of the machine precision and is therefore considered zero. Recalling that the number of non-zero singular values corresponds to the rank of the matrix (the measurement matrix in this case), when $\theta = \pi/3$, the measurement matrix is full-rank. The Penrose inverse can then be used to solve $\mathbf{p} - \mathbf{M}_0 = \mathbf{M}\rho$ and reconstruct the density matrix according to Eq.(19).

(2) New measurements

A new collection of $M = 200$ measurements was carried out. Note that this time $\theta = 5\pi/7$ and that $\mathbf{p}$ was recovered from `ps3.csv`. The vector of singular values is given by

$$\vec{v}_{5\pi/7} = \begin{pmatrix} 8.8183 \\ 7.8369 \\ 7.7987 \end{pmatrix} \tag{29}$$

Therefore the measurement matrix is full rank. The ρ_i, $i \in \{1, 2, 3\}$, coefficients used to reconstruct the density matrix according to Eq.(19) are given by:

Problem 5 [(2) New measurements] continued on next page...

$$\vec{\rho} = \begin{pmatrix} 0.2555 \\ -0.0705 \\ 0.2385 \end{pmatrix} \quad \Longrightarrow \quad \hat{\rho} = \begin{pmatrix} 0.7385 & 0.2555 + 0.0705i \\ 0.2555 - 0.0705i & 0.2615 \end{pmatrix} \tag{30}$$

(3) EWV

It can be checked that the $\hat{\rho}$ matrix obtained is a valid density matrix. Indeed, $\mathrm{Tr}(\hat{\rho}) = 1$ and $\hat{\rho} = \hat{\rho}^\dagger$. However, measurements of probabilities $\mathbf{p}$ are experimental, so there are uncertainties $\mathbf{\Delta p}$ associated with them. These uncertainties will propagate to yield an uncertainty on the reconstructed density matrix. To assess the robustness of the previous tomography procedure, the *equally weighted variance* (EWV) is used

$$EWV = \sum_{n=1}^{3} \sum_{m=1}^{M} (M^+_{nm})^2 (\Delta p_m)^2 \tag{31}$$

where $\mathbf{M}^+$ is the Penrose inverse of the measurement matrix $\mathbf{M}$. Note that the subscript M in the sum corresponds to the number of measurements made ($M = 2001$ in this case) and should not be confused with the measurement matrix M. The uncertainty of the M measurements made in this experiment is estimated as $\Delta p_m = 0.1 + 0.001(m-1)$. Fig.(10) shows the evolution of the quantity EWV as a function of θ, where $\theta \in [0, \pi]$.

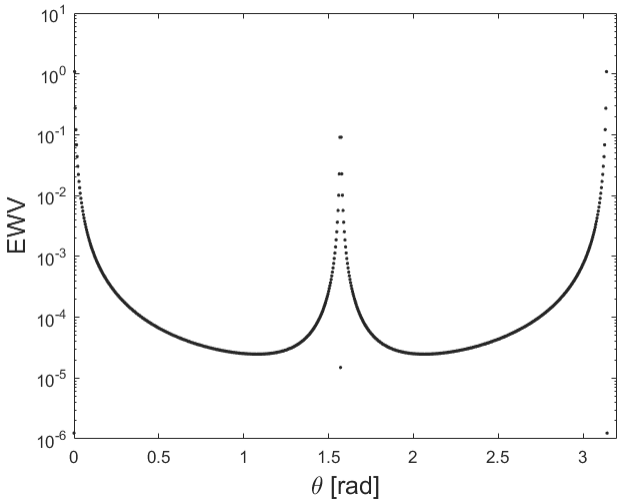


Figure 10: Evolution of EWV as a function of θ, with $\theta \in [0, \pi]$.

Thus, the minima seem to be in $\theta \in \{0, \pi/2, \pi\}$. However, it should be noted that for these values, the $\mathbf{M}$ matrix is singular. As a result, the density matrix cannot be reconstructed from these θ values. Nevertheless, the value of $\theta \approx 1.0477$ also represents a local minimum, for which $\mathbf{M}$ does not have a zero singular value. As a result, the value of θ which minimises the uncertainty $\mathbf{\Delta p}$ and allows the density matrix to be reconstructed is $\theta \approx \pi/3$.