# Quantum information and Computing
# Ex. 06

Andrea Nicolai - 1233407

November 14, 2020

**Abstract**

In this work we were asked to write a program and solve the Schrödinger's equation for the Quantum Harmonic potential. The results were to be analyzed according to some criteria and their coherence with their physical meaning.

## Theory

We are going to solve the Schrödinger's equation given the Hamiltonian operator for 1-dim quantum harmonic oscillator:

$$\hat{H} = \frac{\hat{p}^2}{2m} + \frac{1}{2}\omega^2\hat{x}^2 \tag{1}$$

Where $\hat{p}$ is the momentum operator, $\hat{x}$ is the position operator and the constants $m$ and $\omega$ are respectively the mass of the particle the angular frequency of the oscillator. In the coordinates basis, it rewrites as following:

$$\hat{H} = -\frac{\hbar^2}{2m}\partial_x^2 + \frac{1}{2}\omega^2 x^2 \tag{2}$$

Therefore the Schrödinger's equation to be solved in order to obtain eigenvalues (i.e. the energies $E$) and eigenfunctions $|\psi_E\rangle$ is the following:

$$\hat{H}|\psi_E\rangle = E|\psi_E\rangle \tag{3}$$

From the theory we know that, for a specific energy level $n$ it holds that:

$$E_n = \hbar\omega(1/2 + n) \tag{4}$$

while its eigenfunctions in coordinates basis is:

$$\psi_n(x) = \frac{1}{\sqrt{2^n n!}} \left(\frac{m\omega}{\pi\hbar}\right)^{\frac{1}{4}} e^{-\frac{m\omega x^2}{2\hbar}} \cdot H_n\left(\sqrt{\frac{m\omega}{\hbar}}\right) \tag{5}$$

and $H_n$ are known as *Hermite polynomials*:

$$H_n(z) = (-1)^n e^{z^2} \frac{d^n}{dz^n} \left( e^{-z^2} \right) \tag{6}$$

Note as we are introducing a typical length $x_0 = \sqrt{\frac{\hbar}{m\omega}}$ in our problem that sets bounds for our energy $E$ to be consistent:

$$\frac{\hbar^2}{2mL^2} = E_{IR} \leqslant E \leqslant E_{UV} = \frac{\hbar^2}{2m(dh)^2} \tag{7}$$

where $dh$ is the spacing of our lattice and $L$ is the length of the system. Moreover, it gives us some bounds for energetic levels to be consistent:

$$\frac{x_0^2}{2L^2} = n_{IR} \leqslant n \leqslant n_{UV} = \frac{\hbar^2}{2(dh)^2} \tag{8}$$

# Code Development

The main program to be called accepts as input 5 arguments: length of the system $L$, the resolution $dh$ i.e. lattice spacing, the mass $m$, $\omega$ as defined above and the number $k$ of eigenvectors/eigenfunctions to write in output file. Therefore the total length of the interval will be divided into a certain number of length $dh$. Note that, in order to exploit the symmetry of the system, we centered our total interval in the origin in the coordinates space.
Firstly, we needed to discretize our problem, in particular the second derivative when computing the momentum operator in coordinates space. We defined the second derivative in $x_k$ as follows:

$$\frac{d^2\psi}{dx^2}(x_k) = \frac{\psi_{k+1} - 2\psi_k + \psi_{k-1}}{dh^2} + O(dh^2) \tag{9}$$

This leads to the Laplacian tridiagonal matrix $L_{ij}$ of the kind:

$$L_{ij} = -2\delta_{ij} + \delta_{i,j\pm1} \tag{10}$$

Which is initialized using the module written specifically for complex matrices. Note we have chosen our solution to be null at boundaries, that is to say we have set Dirichlet boundary conditions.
Secondly, using the same module, we introduced another matrix $V_{ij}$ that would stand for the potential. Its only non null terms are the diagonal ones and the ones of the kind:

$$V_{ij} = \frac{1}{2} m(\omega x_i)^2 \delta_{ij} \tag{11}$$

where $x_i$ is the distance from the origin in units of $dh$, that is where the minimum of the potential resides. We expect it there to be null. In addition, in order to exploit the symmetry of the potential, we therefore introduced a check over the main diagonal that would control whether elements are equal, starting from the

central point.
After then, the $ij$-th element of the discretized Hamiltonian matrix would be the sum of the $K_{ij}$, kinetic matrix (i.e. a matrix $L_{ij}$ multiplied by the scalar $-\frac{\hbar^2}{2m(dh)^2}$), and the aforementioned $V_{ij}$:

$$H_{ij} = K_{ij} + V_{ij} = -\frac{\hbar^2}{2m(dh)^2} L_{ij} + V_{ij} \tag{12}$$

Since we know that $H$ is Hermitian, we introduced a checkpoint to verify whether the matrix is selfadjoint. Once all these checks have been passed, we call the subroutine we implemented during the last homework in order to diagonalize the matrix and compute its $2nn + 1$ eigenvalues and eigenvectors. Later, we print the first $k$ for both of them into two different *.dat* files, named according to the specific choice of parameters we used during the run of the program.
Last point to stress out is that the subroutine we call, taken from $-llapack$ package, returns the eigenvectors whose norm squared is equal to one. In other words, the eigenvectors are the ones we would obtain by considering the integral of the norm squared equal to 1 and using all intervals of size 1:

$$\int_L \|\phi_n\|^2 = 1 \tag{13}$$

This is why, when writing a number $num$ of eigenvectors to the output file, we can ask the program to normalize wrt a different size of the intervals that is $dh$, and so be able to write the properly normalized set of $\psi$:

$$\int_L \|\phi_n\|^2 \, dh = 1 \implies \psi_n(x_i) = \frac{\phi_n(x_i)}{\sqrt{dh}} \tag{14}$$

by dividing by $\sqrt{dh}$. Note that, since we know from theory that the only non null eigenvector coefficients are the real ones, we can take into account only them when writing to an output file in order to save space on disk.
All the outputs were finally fed to a Python script to deal with them, for the sake of simplicity and due to the complexity of analysis that would require much more time to be done on gnuplot.

# Results

We run our program by first fixing the length of the interval $L = 0.2$ and $\omega = 5 \cdot 10^4$, $m = 1$, and take into account that we use natural units $\hbar = 1$. The only parameter we have chosen to change, for a matter of time, is the spacing $dh = \{0.05, 0.01, 0.005, 0.001, 0.0005, 0.00025\}$.
As one can see, the typical length of the system is $x_0 \sim 4 \cdot 10^{-3}$, and it holds that $x_0 << L$. Moreover, this was done to not trespass the infrared lowerbound for the energetic levels and make the ground state energy we found consistent.

These limits are $E_{IR} = 50,\ n_{ER} = 0$. In other words, all the eigenvalues we will find must be larger or equal than $E_{IR}$. In addition, it must be taken in account that the UV upperbound as we vary $dh$.

As we expect, increasing the resolution leads to better results and allows us to take into account higher level in energies. This is indeed coherent with what we stated above. As an example, we plot the relative error (see fig. 1) of the energy eigenvalue for different spacings choice. Indeed we can note the presence of the UV limit with regards to the energy levels: for lower energies, and specially for small spacings, the error we make is small. Whereas the more we increase the energy level the higher it becomes. For some choice of spacings it even diverges and values are meaningless. We actually kept them to check what happens when we overcome some limits that our system sets for us.

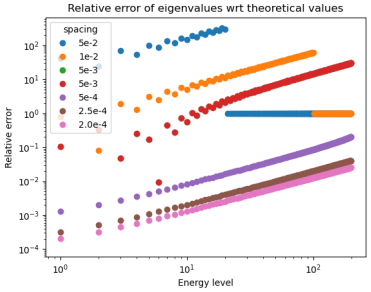
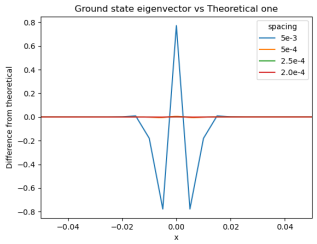
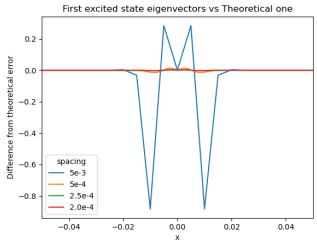



Figure 1: Relative error of the different energy levels and for different choice of spacings wrt theoretical ones.

We now compare the first two eigenfunctions (figg. 2a, 2b), in particular the ground state one and the first excited state. We decided to plot the difference between the absolute value of the "computational" one and the theoretical one, and not the relative error since it might have lead to some numerical instability when the wavefunction is were null. We took the absolute value in order to take into account that the program may return either $\psi_n$ or $-\psi_n$, being both solutions of the differential equation. As we expected, the higher in resolution, the better the theoretical curve is approximated: moreover we can note as we tend to underestimate the wavefunction at its boundaries, while we tend to overestimate it. This can be due to discretization.

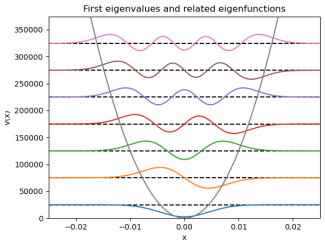



(a) Difference between the two absolute value of the eigenfunctions computed for different spacings and theoretical **ground state**.

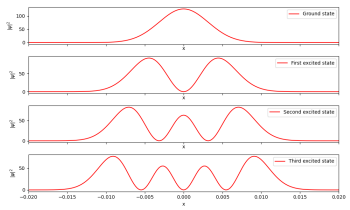



(b) Difference between the two absolute value of the eigenfunctions computed for different spacings and theoretical **first excited state**.

Now we want to take a qualitative overview over the eigenfunctions we obtained: we want to check whether the parity is the one we would expect. As it is shown in fig 3a, one can see that for even $n$, the related $\psi_n$ is even too. The converse is true as well: odd $n$ lead to odd eigenfunctions. Moreover, for the norm plot 3b the symmetry of the system, which is given by the symmetric interval and potentials, is preserved as well.




(a) Potential and first 7 eigenfunctions, which are NOT in scale, are shown here. Horizontal lines are the energetic levels.




(b) Norm of the first 3 eigenfunctions. Their integral is 1.

Figure 3

## Self-evaluation

- **Correctness**: Results are actually coherent with the ones we would expect from theory up to a certain precision, and some checkpoints were implemented in order to be sure of that intermediate steps would be correct.
- **Stability**: as one can note, sometimes we divide by either the spacing or the square of the spacing. He should keep in mind that, if the spacing is too little, it can turn out to be a problem. However, if the spacing is too large, we may end up with some overflow errors as in 1.
- **Accurate discretization**: results are consistent with the theory, so up to a certain spacing we can be satisfied with our discretization. However, the limits we briefly introduced when dealing with the theory must be taken into account when setting a spacing value.
- **Flexibility**: the code can be used by varying some of the parameters we kept fixed, as an example $\omega$. However, one should take into account that the program was written exploiting the symmetry of the harmonic potential wrt the origin.
- **Efficiency**: unfortunately, diagonalizing the matrix requires some time, and the larger the matrix, the less efficient the program would be.