Week 6: Continuous Time-Independent Schrödinger Equation

Alice Pagano

(Dated: November 17, 2020)

In this Report, we solve a quantum harmonic oscillator system. In particular, we compute eigenvalues and eigenvectors of quantum harmonic hamiltonian by discretize it with finite difference method. At the end, we compare the obtained numerical results with the analytical ones.

I. THEORY

A. Quantum Harmonic Oscillator

Let us consider the **one-dimensional quantum harmonic oscillator** defined by the Hamiltonian:

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

where m is the particle's **mass**, $\omega = \sqrt{k/m}$ is the **angular frequency**, $\hat{x}$ is the **position operator** and $\hat{p} = -i\hbar\frac{\partial}{\partial x}$ is the **momentum operator**. In particular, the first term in the Hamiltonian represents the *kinetic energy* of the particle, and the second term represents its *potential energy*.

One may write the **time-independent Schrödinger equation** for a quantum harmonic oscillator as:

$$\hat{H}\left|\psi_n\right\rangle = \left(\frac{\hat{p}^2}{2m} + \frac{1}{2}m\omega^2\hat{x}^2\right)\left|\psi_n\right\rangle = E_n\left|\psi_n\right\rangle \tag{2}$$

In *coordinate basis*, there is a family of solusions which amount to **Hermite functions**:

$$\psi_n(x) = \frac{1}{\sqrt{2^n n!}}\Big(\frac{m\omega}{\pi\hbar}\Big)^{1/4} e^{-\frac{m\omega x^2}{2\hbar}} H_n\left(\sqrt{\frac{m\omega}{\hbar}}x\right), \qquad n = 0, 1, 2, \dots \tag{3}$$

where the Hermite polynomials are:

$$H_n(x) = (-1)^n e^{z^2}\frac{\mathrm{d}^n}{\mathrm{d}z^n}\left(e^{-z^2}\right) \tag{4}$$

The corresponding **energy levels** are:

$$E_n = \hbar\omega\left(n + \frac{1}{2}\right) \tag{5}$$

The first four eigenfunctions are:

$$\psi_0(x) = c_0 e^{-\frac{mx^2\omega}{2\hbar}} \tag{6a}$$

$$\psi_1(x) = c_0\sqrt{2}\sqrt{\frac{m\omega}{\hbar}}x e^{-\frac{mx^2\omega}{2\hbar}} \tag{6b}$$

$$\psi_2(x) = \frac{c_0}{\sqrt{2}}\left(\frac{2mx^2\omega}{\hbar} - 1\right)e^{-\frac{mx^2\omega}{2\hbar}} \tag{6c}$$

$$\psi_3(x) = \frac{c_0}{\sqrt{3}}\left(2\Big(\frac{m\omega}{\hbar}\Big)^{3/2}x^3 - 3\sqrt{\frac{m\omega}{\hbar}}x\right)e^{-\frac{mx^2\omega}{2\hbar}} \tag{6d}$$

where $c_0 = \big(\frac{m\omega}{\hbar\pi}\big)^{1/4}$.

B. Finite Differences Method

For the sake of simplicity, from now let us impose $m = 1$, $\hbar = 1$ and $\omega = 1$; the Schrödinger equation of the system becomes:

$$\frac{1}{2}\left(\hat{p}^2 + \hat{x}^2\right)|\psi_n\rangle = E_n\,|\psi_n\rangle \tag{7}$$

with eigenvalues $E_n = n + \frac{1}{2}$. To solve numerically this equation, one can introduce artificially a lattice with constant spacing h in each spatial direction. In particular, we fix an inteval $[a : b]$ and we divide it in N parts such that $h = (b-a)/N$. By using the *finite differences method*, the **discrete** version of the time-independent Schrödinger equation, at the *fourth order*, for our harmonic oscillator in a one dimensional interval $[a : b]$, is given by:

$$\begin{pmatrix} \frac{1}{h^2}+\frac{x_0^2}{2} & -\frac{1}{2h^2} & 0 & \dots & 0 & 0 & 0 \\ -\frac{1}{2h^2} & \frac{1}{h^2}+\frac{x_1^2}{2} & -\frac{1}{2h^2} & \dots & 0 & 0 & 0 \\ & & & \vdots & & & \\ 0 & 0 & 0 & \dots & -\frac{1}{2h^2} & \frac{1}{h^2}+\frac{x_{N-1}^2}{2} & -\frac{1}{2h^2} \\ 0 & 0 & 0 & \dots & 0 & -\frac{1}{2h^2} & \frac{1}{h^2}+\frac{x_N^2}{2} \end{pmatrix} |\psi_n\rangle = E_n\,|\psi_n\rangle \tag{8}$$

where $x_i = a + ih$.

II. CODE DEVELOPMENT

In order to solve the quantum harmonic oscillator system, i.e. finding eigenvalues and eigenvectors of its hamiltonian operator, we develop a program inside the file "hermitian.f90". The main steps of the program are:

1. the space interval [`min` : `max`] and the number of points in which subdivide the system `N`, are taken as input;
2. the discretization step `h` is computed as (`max-min`)/`N`;
3. the hamiltonian of a quantum harmonic oscillator, is discretized with *finite difference method* and the resulting matrix is **tridiagonal**. To store it in an efficient way, we initialize a vector `Ad` with its *diagonal* part and a vector `Asubd` with its *subdiagonal*;

```
! initialize hermitian matrix A
do ii=0,N
    Ad(ii) = 1/(h**2) + 0.5*(min + ii*h)
    **2
    if(ii<N) then
        Asubd(ii) = -1/(2*h**2)
    end if
end do
```

4. then, the eigenvalues and eigefunctions are computed by calling the Lapack `SUBROUTINE` **`dstev`**. In particular, after the calling, the vector `Ad` will contain the matrix eigenvalues, while the array `A` will have as columns the eigenfunctions;

```
! compute eigenvalues and eigefunctions
call dstev('V', N+1, Ad, Asubd, A, N+1, WORK, info)
```

5. eigenfunctions are normalized being multiplied by the factor $\sqrt{\frac{N+1}{2\,max}}$;
6. at the end, eigenvalues and eigenfunctions are printed into different files.

Then, by using a python script "script.py", we fix `min`, `max` and `N` variables and we run the main program. After that, eigenfunctions, up to a fixed `Nmax` order, are plotted with a gnuplot script "plot_eig_func.p". Then, the difference between numerical and analytical eigenvalues is plotted by using another script called "plot_eig_val.p".

III. RESULTS

We run the program for `N=1000` and for different space intervals. In Fig. 1, we plot the first four eigenfunctions for $[-3,3]$ and $[-5,5]$ and we compare the numerical results for the two different ranges with the analyticial ones. We can see how the lower order numerical eigenfunctions in the range $[-5,5]$ are in quite accordance with the theory. However, by increasing the eigenfunction's order and in particular after the 8th eigenfunction they start to differ. Instead, the numerical results in $[-3,3]$ slightly differs since the beginning and differs even more by increasing the eigenfunction's order. This is due to the fact that the choosen range should be in accordance with the physical parameters fixed, i.e. $\omega = 1$. This problem is like the one of a particle moving in a potential well: if we consider a too small range as $[-3,3]$, we will have boundary effects which will spoil our result. By our analysis, we can conclude that the best range we can choose for such a problem is $[-5,5]$.

We also analyze the difference between the numerical eigenvalues obtained by the simulation and the analytical ones. The results are plotted in Fig. 2 for both $[-3,3]$ and $[-5,5]$. We can see that:

- for $n < 10$, the difference is very little for eigenvalues calculated in $[-5,5]$, while is quite large for the ones in $[-3,3]$;
- for $n > 10$, both for $[-5,5]$ and $[-3,3]$ the difference diverges.

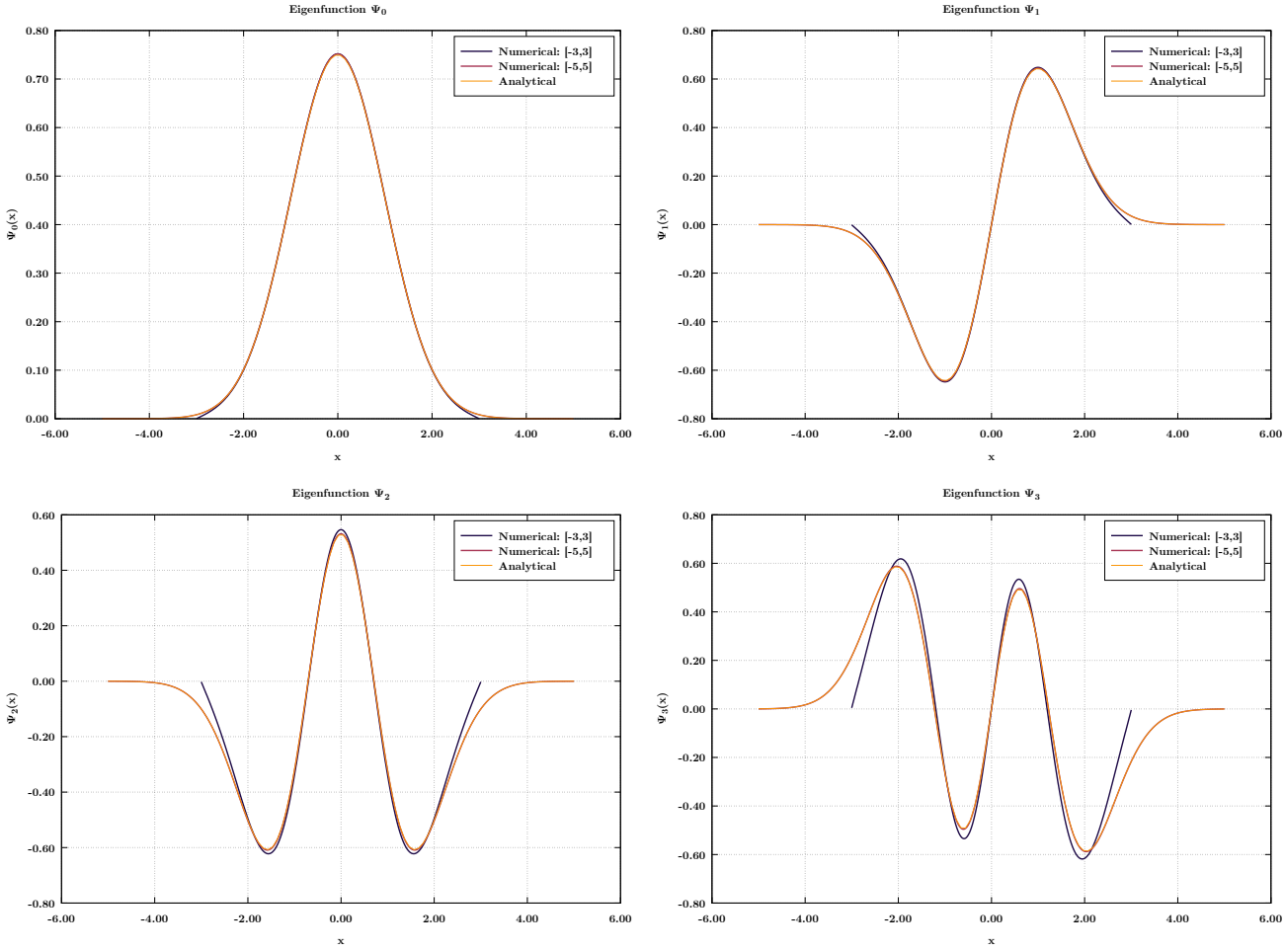

FIG. 1 Plot of first four numerical eigenfunctions for a quantum harmonic oscillator with a space interval of $[-3,3]$ and $[-5,5]$. The numerical functions are compared with the analytical ones.

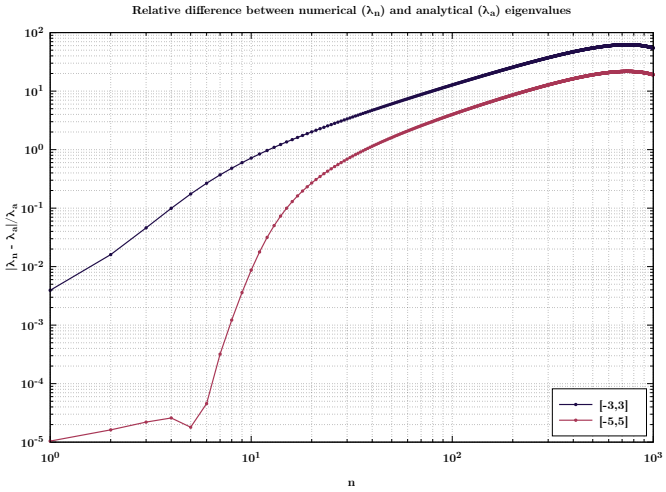

FIG. 2 Plot of relative difference between numerical and analytical eigenvalues for $[-3, 3]$ and $[-5, 5]$ space intervals.

IV. SELF-EVALUATION

Let us rate our program in terms of the priorities for good scientific software development:

- **correctness**: if we consider lower degree eigenfunctions, for a proper choice of space range, the obtained results perfectly match the theoretical one. Hence, the code is correct but still limited by the discretization and range we used;
- **stability**: the code is numerically stable since it returns the same results by running it several times and with several paramters;
- **accurate discretization**: accurancy can be further improved by choosing a lower discretization step;
- **flexibility**: for the sake of simplicity, we impose $\omega = 1, \hbar = 1, m = 1$ in the main program; however, the program can be easily adapated for other values of the parameter for the quantum harmonic oscillator system;
- **efficiency**: the program is quite efficient and run in few seconds if we choose a slice dimension of $N = 1000$ or also $N = 2000$. Indeed, in the main program we exploit the fact that the hamiltonian is tridiagonal by finding eigenvalues and eigenvector with the specialized Lapack's function for tridiagonal matrices which are faster than generic ones.