Assignment 4

Quantum Information and Computing Course 2022/2023

Massimo Colombo

Department of Physics and Astronomy "Galileo Galilei"
DEGREE IN PHYSICS

November 29, 2022

Time-independent Schrodinger Equation: Theory

The following Schrödinger equation corresponds to the quantum harmonic oscillator's system (with all physical constants set to 1):

$$\hat{H}\,|\Psi> \;=\; E\,|\Psi> \quad with \quad \hat{H} \;=\; \hat{p}^2 \;+\; \omega^2\,\hat{q}^2$$

$$E_n \;=\; 2n \;+\; 1, \qquad\qquad |\Psi_n> \;=\; A_n \; exp\left(-\,\omega\,\frac{x^2}{2}\right)\; H_n(x)$$

where A_n are normalization constants and H_n are the Hermite polynomials.
E_n and $|\Psi_n>$ are the eigenvalues and eigenvectors analytical solutions.

Within a computational approach, the steps to solve the problem are:

- To discretize the system, considering a finite range $[x_{min}, x_{max}]$ in the x coordinate, with N steps of length dx ($= (x_{max} - x_{min})/N$). This procedure defines a 1-dim lattice $\{x_{min}, x_{min} + dx, ..., x_{max}\}$ where the numerical eigenvectors will be defined.
- To discretize the Hamiltonian defined upon the lattice and to diagonalize it, obtaining eigenvalues and eigenvectors.

Schrodinger Equation: Code Development

In coordinate's basis, the Hamiltonian presents a 2^{nd} derivative operator $\hat{p}^2 \longrightarrow -\frac{d^2}{dx^2}$ and the squared position operator $\hat{q}^2 \longrightarrow x^2$.

Remembering the approximation: $f_n'' = \frac{f_{n+1} - 2f_n + f_{n-1}}{dx^2} + O(dx^2)$, we obtain the following matrix:

$$\begin{pmatrix} \frac{2}{dx^2} + x^2 & -\frac{1}{dx^2} & 0 & \dots & 0 & 0 \\ -\frac{1}{dx^2} & \frac{2}{dx^2} + x^2 & -\frac{1}{dx^2} & \dots & 0 & 0 \\ 0 & -\frac{1}{dx^2} & \frac{2}{dx^2} + x^2 & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \frac{2}{dx^2} + x^2 & -\frac{1}{dx^2} \\ 0 & 0 & 0 & \dots & -\frac{1}{dx^2} & \frac{2}{dx^2} + x^2 \end{pmatrix} |\Psi> = E\,|\Psi>$$

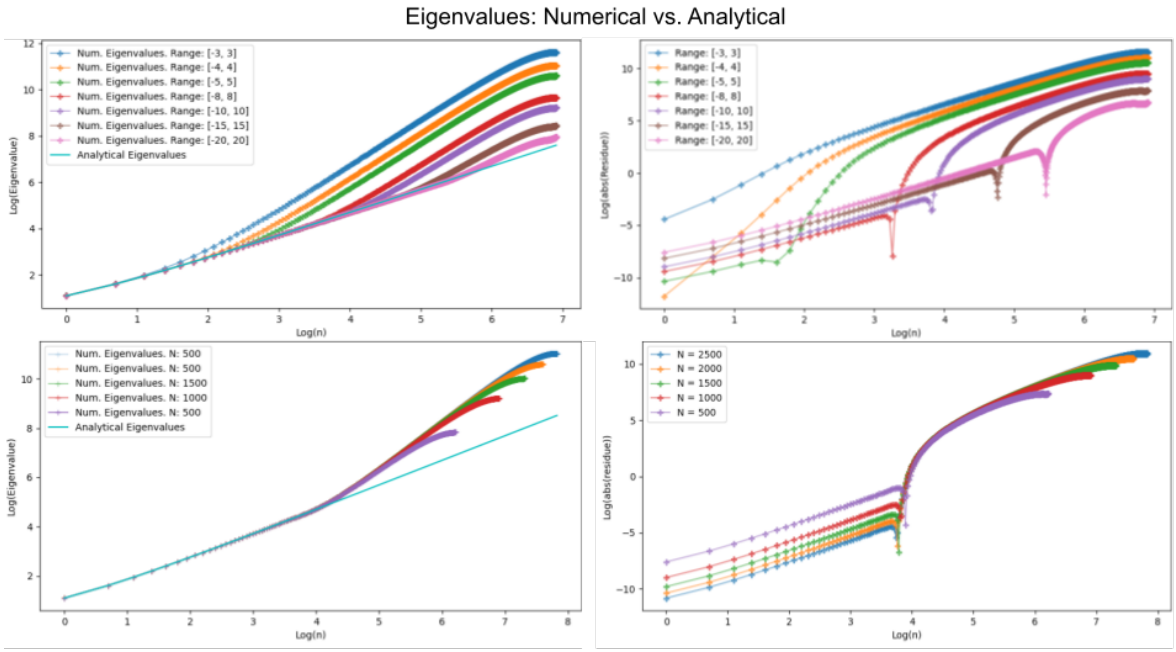
```
!lattice definition
dx = (x_max - x_min) / (NN - 1)
DO ii = 1, NN
    ax_grid(ii) = x_min + (ii - 1) * dx
END DO
!define tridiagonal matrix
DO ii = 1, NN
    diagonal(ii) = 2/dx**2 + (omega * ax_grid(ii))**2
    IF (ii /= NN) THEN
    subdiagonal(ii) = -1/dx**2
    END IF
END DO
CALL dsteqr("I", NN, diagonal, subdiagonal, eigenvectors, NN, WORK, INFO)
```

This *tridiagonal* matrix has been diagonalized thanks to the LAPACK routine **dsteqr()**. The input "I" tells the routine to return both eigenvalues and eigenvectors.

Thanks to the module *"command line args"*, it was possible to run the program in debug mode and script mode:
STDIN: ./Ex1.out –debug0 ./Ex1.out –script
This was helpful for the analysis.

Time-independent Schrodinger Equation: Results

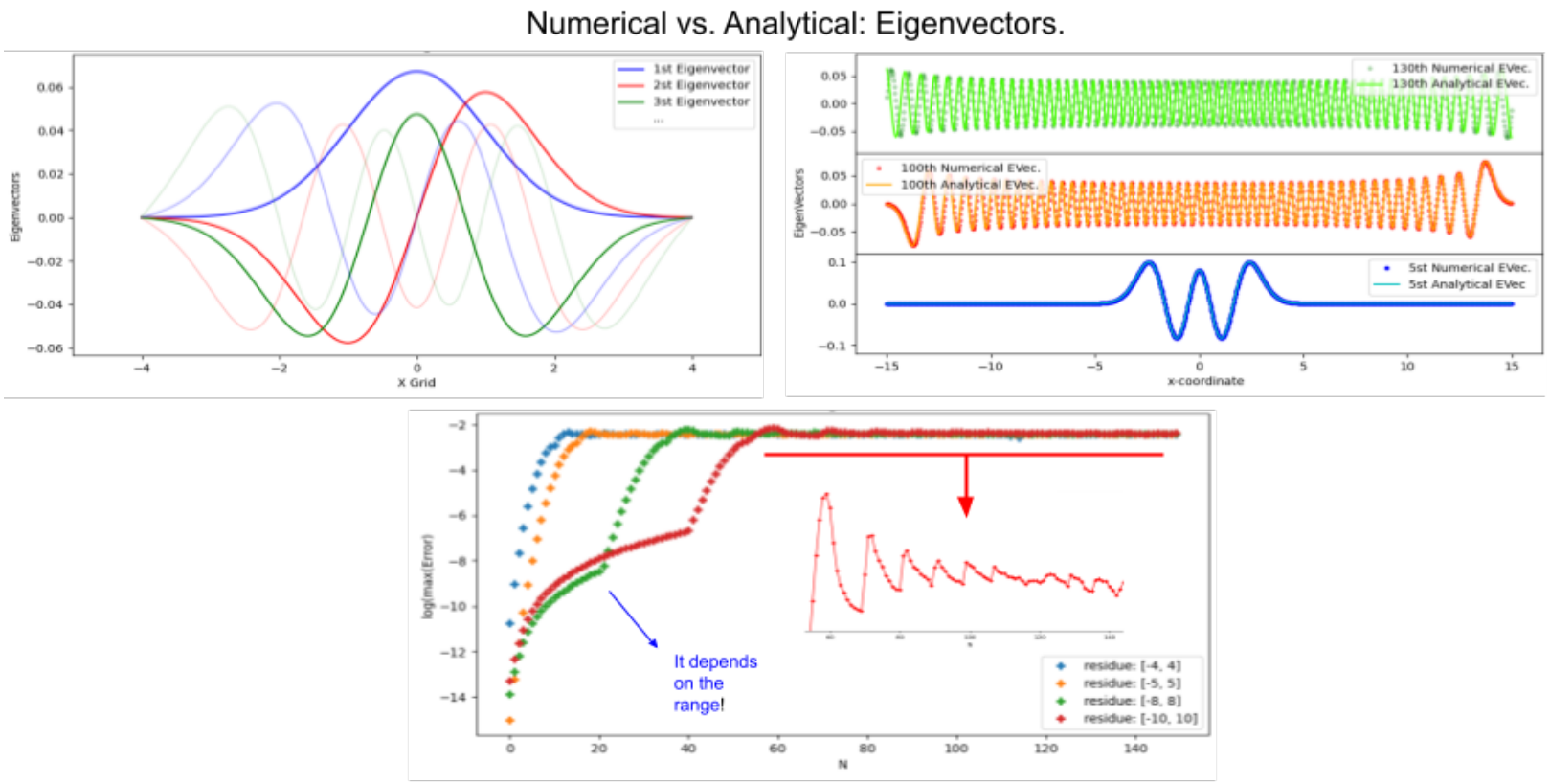
In the following, the comparison between numerical and analytical eigenvalues is presented. To check code's behaviour, different ranges $[x_{min}, x_{max}]$ and numbers of step "N" have been studied:



From the 1st plot, we can see that, at some order n (for each chosen range), the numerical eigenvalues diverge from the analytical ones. Meanwhile, from the 4th plot, increasing the number of steps "N" reduces their difference with respect to the same eigenvalue, as expected with a better lattice discretization.

Time-independent Schrodinger Equation: Result 2

The same has been done analyzing Eigenvectors:



After checking the correct shape (1st plot), a phase displacement is seen when comparing higher-order eigenvectors (plot 2). This leads to a constant maximum difference that is about two times the amplitude of the n-th eigenvalue, as shown in plot 3. Moreover, with better discretization, the difference between analytical and numerical results becomes lower, as before.

Overall Insight

- The previous analysis shows the compatibility for the first n eigenvalues and eigenvectors of the numerical results with the theoretical ones. This increases the confidence in the correctness of the program.

- The code checks STDINs to be coherent and meaningful for the algorithm.
For instance:

STDOUT: *Enter the number of points you want to discretize the axis interval with: (integer positive value)*
STDIN: 1.5
STDOUT: *!IOSTAT > 0, the previous READ has encountered some problems.*
A common problem is illegal data: supplying a real number to an integer variable

All the "IF conditions" and "type-data checking" are utilized to increase stability.

- An accurate discretization seems to have been adopted. The approach of the numerical results to the theoretical ones is shown in the analysis when utilizing a denser lattice to solve the problem.

- To conclude, the program accepts STDIN values for x_{min}, x_{man}, and N. So, it can analyze a wide interval of ranges and discretizations. However, the code diagonalizes only *tridiagonal matrices*. So, from that point of view, it lacks in flexibility, while gaining in efficiency.