Computational Physics II — Project 5

Jan O. Haerter

April 27, 2022

Hand-in date: 6 a.m., Friday, May 13, 2022

In total, **100 pts** can be achieved by solving the problems below (see annotation).

---

1 Stationary (an)harmonic oscillator

Consider the quantum harmonic oscillator

$$\frac{1}{2}(-\partial_u^2 + u^2)\, v(u) = \varepsilon\, v(u)\,, \tag{1}$$

with variables as defined in class. The symbols ε, v and u, instead of E, Ψ and x, are used to indicate the dimensionless formulation.

1. Using the so-called "shooting method" (as discussed) determine the lowest 10 eigenvalues and corresponding normalized eigenstates. (*Hint:* It is important to start the integration at a value of u where the wavefunction is assumed sufficiently close to zero, yet, too large values of $|u|$ will make the integration numerically difficult due to excessive divergence. Some trial and error, or a sophisticated guess will help.) (20pts)
2. Plot the wavefunction $v(u)$ vs. u for all wavefunctions $v(u)$ obtained, and label them appropriately. Additionally, plot the first few analytically obtained eigenfunctions (Hermite polynomials) to assess your numerical results. (15 pts)
3. allow for an additional term $\propto \alpha\, u^4\, v(u)$ (a fourth-order potential) on the LHS of Eq. 1, yielding an anharmonic oscillator. For a suitable value of α again obtain the first 10 eigenfunctions and corresponding eigenvalues. Discuss the difference to those obtained from the harmonic oscillator. (15 pts)

2 Propagating wavepacket

Consider the Gaussian wavepacket

$$v(u,0) = \left(\frac{1}{\pi}\right)^{1/4} e^{-u^2/2}\,, \tag{2}$$

discussed in class, that is, the ground state of the 1D harmonic oscillator. (10 pts each below).

1. Discretize space and represent $v(u,0)$ as a discrete function of space.

2. Using an appropriate discrete (fast) Fourier transform, transform Ψ to momentum space and act on it with the free-particle ($V(u) = 0$) time evolution operator, to obtain $v(u, \Delta t)$ for a small value of Δt. Repeat for a sufficient number n of timesteps to propagate to $v(u, n\Delta t)$. Plot $v(u, m\Delta t)$ for several values of m, check the normalization and compare with the exact results obtained in class.

3. Give the wavepacket an initial momentum k_0 by multiplying Eq. 2 by a phase factor $e^{-ik_0 u}$. Introduce a potential barrier $V(u) = V_0$ for suitably chosen $0 < u_1 \leq u \leq u_2$ and $V(u) = 0$ otherwise. Allow yourself to modify the constant V_0 and discuss the behavior at the interfaces $u = u_1$ and $u = u_2$. (*Hint*: To incorporate the potential $V(u)$ into your code, it is recommended to apply the potential in position space and the kinetic energy operator in momentum space.)

4. Finally, return to the harmonic oscillator potential $V(u) = au^2$ and allow the wavepacket to propagate within this potential. Compute the position expectation value $u(t) \equiv \langle v(u,t)|\hat{u}|v(u,t)\rangle$ for each timestep and plot $u(t)$ vs t. Similarly, obtain the momentum expectation value $k(t) = \langle \Psi(k,t)|\hat{k}|\Psi(k,t)\rangle$ and add it to the plot. Qualitatively discuss your findings in relation to classical expectations.

5. Document your results by appropriate figures or animations (the latter is quite straightforward to do in Python using `animation.FuncAnimation` and may be most useful).

General remarks for all Projects. You will have to (i) analyze the problem, (ii) select an algorithm (if not specified), (iii) write a Python program, (iv) run the program, (v) visualize the data numerical data, and (vi) extract an answer to the physics question from the data. Which checks did you perform to validate the code? State the results you got for these tests. For each project you will submit a short report describing the physics problem, your way of attacking it, and the results you obtained. Provide the documented Python code in such a form that we can run the code. A Jupyter Notebook including the code and report is fine but not necessary.