Quantum Oscillator HW 1: Using a shooting method to solve the ODE

Computational Goal: Find first 5 eigen values/functions $\boldsymbol{\varepsilon}_n$ for the second order ODE $\frac{d^2\phi_n}{dx^2} - [Kx^2 - \boldsymbol{\varepsilon}_n]\phi_n = 0$ with $K = 1$, using the shooting method.

1 How do I approximate the problem on a finite domain, what would be the boundary conditions?

Implementing the Boundary Conditions requires some approximations which have not been stated explicitly in the problem statement, hence are discussed in detail here.
Consider the simplified second-order differential equation:

$$\frac{d^2\phi_n}{dx^2} - \beta_n\phi_n = 0 \tag{1}$$

with same boundary conditions $\phi_n(x) \to 0$ as $x \to \pm\infty$. This system has the general solution:

$$\phi_n(x) = c_1 \exp(\sqrt{\beta_n}x) + c_2 \exp(-\sqrt{\beta_n}x) \tag{2}$$

where only $\beta_n \geq 0$ are considered to have decaying solutions $\phi_n(x)$. Furthermore, for decaying solutions we require:

$$x \to \infty : \phi_n(x) = c_2 \exp(-\sqrt{\beta_n}x) \tag{3}$$

$$x \to -\infty : \phi_n(x) = c_1 \exp(\sqrt{\beta_n}x) \tag{4}$$

These solutions can be also expressed as the solutions to the first order equations:

$$x \to \infty : \frac{d\phi_n}{dx} + \sqrt{\beta_n}\phi_n = 0 \tag{5}$$

$$x \to -\infty : \frac{d\phi_n}{dx} - \sqrt{\beta_n}\phi_n = 0 \tag{6}$$

Comparing the above equations, with the constant β_n value replaced by the variable $\beta_n(x) = Kx^2 - \varepsilon_n$, a good approximation to implement the required boundary conditions $\phi_n(x) \to 0$ as $x \to \pm L$ at the boundaries of the computational domain $x \in [-L, L]$ is given by equations:

$$x = L : \frac{d\phi_n}{dx} + \sqrt{Kx^2 - \varepsilon_n}\phi_n = 0 \tag{7}$$

$$x = -L : \frac{d\phi_n}{dx} - \sqrt{Kx^2 - \varepsilon_n}\phi_n = 0 \tag{8}$$

To get an idea of the error introduced due to the above approximation, computing the derivative of the boundary condition equations with $K = 1$ gives:

$$\left.\frac{d^2\phi_n}{dx^2}\right|_{\pm L} = \left(\frac{\pm x}{\sqrt{x^2 - \varepsilon_n}}\phi_n + \left(x^2 - \varepsilon_n\right)\phi_n\right) \tag{9}$$

Hence the error introduced is $\frac{\pm x}{\sqrt{x^2 - \varepsilon_n}}\phi_n$, which is small as $\phi_n(x) \to 0$ as $x \to \pm L$.

2 Algorithm sketch

We have to solve a "boundary value problem", which requires that all the solutions of the differential equation satisfy two conditions prescribed above. You shooting method/algorithm to solve such a boundary value problem should:

1. For every eigenfunction, eigenvalue pair $[\phi_n, \varepsilon_n]$ to be computed, start with an initial guess for ε_n and $\phi_n \mid_{-L} = \phi(-L)$. Hint: The theoretical eigen values are odd integer values $\varepsilon_n = 1, 3, \cdots$. Since the ODE is **linear**, the initial guess value of $\phi_n(-L)$ can be arbitrary, for convenience it can be chosen as 1.

2. Solve the problem using the MATLAB function `ode45`, giving values at the grid points $x_n = -L \ : \triangle x \ : \ L$ of ϕ_n.

3. If solution at $x = L$: $\phi_n(L)$ satisfies the prescribed boundary condition within required tolerance, the required eigenvalue ε_n have been found. If the solution does not meet the required boundary condition $x = L$, then adjust the value of ε_n in **progressively smaller** step sizes till the solution converges. For instance, you can keep cutting the amount by which you change ε_n by half. Remember to reset this for each eigen value.

3 Other possible approximations for the boundary conditions?

The boundary conditions prescribed above is just one possible approximation - this is the one set up for the grader. As mentioned in the general FAQ, the grader is in place as a check on your algorithm steps. Once your shooting algorithm produces the same results as the grader, you know that you have implemented the algorithm correctly, awesome! Feel free to play around with other approximations/domain sizes to see how your eigen values compare to the theoretical eigen values.