# Scientific Computing

Department of Applied Mathematics
University of Washington

# Outline

Homework problem:

$$\frac{d^2\phi_n}{dx^2} - [Kx^2 - \epsilon_n]\phi_n = 0 \qquad (1)$$

subject to BC:
$\phi_n(x) \to 0$ as $x \to \pm\infty$

The first thing we need to do about this 2nd order ODE is to convert it into a system of 1st orde ODEs which can be solved by ODE45. Let $y_1(x) = \phi_n(x)$ and $y_2(x) = \frac{d\phi_n(x)}{dx}$ and plug into equation (5), we get:

$$\begin{cases} \frac{dy_1}{dx} = y_2 \\ \frac{dy_2}{dx} = (Kx^2 - \epsilon_n)y_1 \end{cases} \tag{2}$$

Now the original equation is in the form that can be easily solved by ODE45, however, there are still two things we need to figure out:

1. How to compute the value of $\epsilon_n$?
2. What ICs should be used?

For our first concern, we can use the shooting method to find the proper $\epsilon_n$ which should be similar as we did in lecture notes ( in lecture notes, we shoot for the boundary condition).

For the second concern, we can actually consider two conditions which is equivalent to the original boundary conditions from the numerical point of view:

$$\frac{d\phi_n(-L)}{dx} - \sqrt{KL^2 - \epsilon_n}\phi_n(-L) = 0 \tag{3}$$

$$\frac{d\phi_n(L)}{dx} + \sqrt{KL^2 - \epsilon_n}\phi_n(L) = 0 \tag{4}$$

If we move the second term of (3) to the RHS, we can easily get

$$\frac{d\phi_n(-L)}{dx} = \sqrt{KL^2 - \epsilon_n}\phi_n(-L)$$

which tells us given

$$\phi_n(-L) = A$$

, one can get

$$\frac{\phi_n(-L)}{dx} = A\sqrt{KL^2 - \epsilon_n}$$

But one might ask, what value we should pick for $A$, the answer is $A$ can be any constant. Why? If you are careful enough, you might already noticed, assume $\phi_n$ is a solution to (1), then $c\phi_n$ is also a solution to (1), one can prove this by plug $c\phi_n$ into the equation. Thus, mutiply the IC by some constant won't make us in trouble and we will normalized the solution in the end.

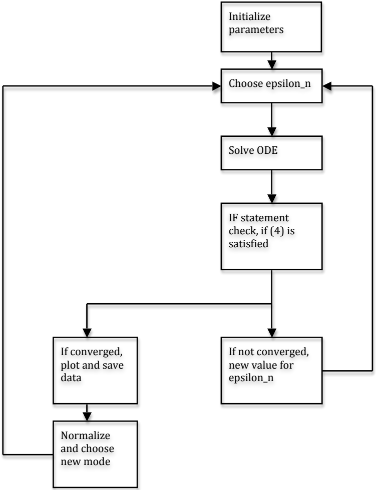


Figure : Flowchart for implementing shooting method

Homework problem:

$$[-\frac{d^2}{dx^2} + Kx^2]\phi_n = \epsilon_n\phi_n \tag{5}$$

subject to BC:
$\phi_n(x) \to 0$ as $x \to \pm\infty$

Still, as we did in Prob1, the original bounary condition is converted to a equivalent form below:

$$\frac{d\phi_n(-L)}{dx} - \sqrt{KL^2 - \epsilon_n}\phi_n(-L) = 0 \tag{6}$$

$$\frac{d\phi_n(L)}{dx} + \sqrt{KL^2 - \epsilon_n}\phi_n(L) = 0 \tag{7}$$

The only difference in this problem is we need to apply direct method to our problem.

Center differnce scheme

$$\frac{d^2}{dx^2}f(x) = [f(x+\Delta x) - 2f(x) + f(x-\Delta x)]/\Delta x^2 \qquad (8)$$

is used to approximate differential operator $\frac{d^2}{dx^2}$, plug it into eq(1), get

$$[\phi(x_{i+1}) - 2\phi(x_i) + \phi(x_{i-1})]/\Delta x^2 - [Kx_i^2 - \epsilon_n]\phi_n(x_i) = 0 \qquad (9)$$

For boundary conditions using forward- and backward- schemes, on left and right bounaries respectively, we can get

$$\frac{d\phi_n(x_0)}{dx} = \sqrt{Kx_0^2 - \epsilon_n}\phi_n(x_0) = [-3\phi_n(x_0)+4\phi_n(x_1)-\phi_n(x_2)]/2\Delta t \tag{10}$$

And omitting the $\Delta t$ term(since very small), get

$$3\phi_n(x_0) = 4\phi_n(x_1) - \phi_n(x_2) \tag{11}$$

Similarly, we can get

$$3\phi_n(x_n) = 4\phi_n(x_{n-1}) - \phi_n(x_{n-2}) \tag{12}$$

Plug eq(7) and (8), in to eq(5) on the boundary such that we can get rid of $\phi(x_0)$ and $\phi(x_N)$.Thus we can reduce the equation as an eigenvalue problem as following

$$A\phi = (\epsilon \Delta x^2)\phi$$

where

$$\begin{pmatrix} \frac{2}{3} + \Delta x^2 K x_1^2 & -\frac{2}{3} & & & \\ -1 & 2 + \Delta x^2 K x_2^2 & -1 & & \\ & & \ddots & & \\ & & -1 & 2 + \Delta x^2 K x_{N-2}^2 & -1 \\ & & & -\frac{2}{3} & \frac{2}{3} + \Delta x^2 K x_{N-1}^2 \end{pmatrix}$$

Notice that the equation in Prob3, has an extra term $\gamma|\phi|^2\phi$, which makes the equation no longer liner, so shooting for $\epsilon_n$ is not enough, we should shoot for $A$(in the initial condition) as well when we are shooting for $\epsilon_n$.Recall that we used the following condition as IC for last HW and $A$ can be arbitrary numbers in Prob1

$$[A, A\sqrt{KL^2 - \epsilon_n}] \tag{13}$$

Since what we want are normalized eigenfunctions, and this time, because of the nolinearity, if we normalize the eigenfunctions by dividing by the squre roots of their norms, the eigenvalues will not correspond to those normalized eigenfunctions, so we need to solve for both normalized eigenfunction and the eigenvalue. Here in Ex3 we can use the normalization as the shooting condition for $A$, i.e. we want

$$abs(\|\phi_n\| - 1) < tol$$

# Problem 3

```
for modes=1:2
    for j=1:1000
        A = A_start;
        for k=1:1000
            update initial condition
            ode45
            compute norm
            if abs(norm - 1) < tol
            else if norm > 1
                A = A-dA;
            else
                A = A+dA/2;
                dA=dA/2;
            end
        end
        epsilon shooting
    end
```

# Alternate Method

Here is another way to shoot for A to enforce $\|\phi_n\| = 1$.

```
for modes=1:2
    for k=1:1000
        update initial condition
        ode45
        compute norm
        if abs(norm-1) < tol
           break
        else
           A = A/sqrt(norm); % normalizes each time
        end
        epsilon shooting
    end
    epsilon_start = epsilon + 0.1;
 end
```