EP547 Computational Methods in QM

Topic 14

Schrödinger Equation

Department of Engineering Physics

University of Gaziantep

May 2013
Eigenvalue Problems

Eigenvalue problems are a special class of boundary value problems that are common in physics and engineering problem contexts involving:

 agli Vibration
agli Elasticity
agli Quantum Mechanics etc.
We have seen that a Linear system equations can be written as in matrix form:

\[
\begin{pmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nn}
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
\vdots \\
x_n
\end{pmatrix}
= 
\begin{pmatrix}
b_1 \\
b_2 \\
\vdots \\
b_n
\end{pmatrix}
\]

or

\[Ax = b\]

For homogeneous systems, we have:

\[Ax = 0\]
An eigenvalue problem is typically expressed as:

\[
\begin{pmatrix}
 a_{11} - \lambda & a_{12} & \cdots & a_{1n} \\
 a_{21} & a_{22} - \lambda & \cdots & a_{2n} \\
 \vdots & \vdots & \ddots & \vdots \\
 a_{n1} & a_{n2} & \cdots & a_{nn} - \lambda \\
\end{pmatrix}
\begin{pmatrix}
 x_1 \\
 x_2 \\
 \vdots \\
 x_n \\
\end{pmatrix} =
\begin{pmatrix}
 0 \\
 0 \\
 \vdots \\
 0 \\
\end{pmatrix}
\]

where \( \lambda \) is known as eigenvalue and \( \mathbf{x} \) as eigenvector. The equation may also be written as

\[
(A - \lambda I) \mathbf{x} = 0
\]

One way to get non-trivial solution is based on getting \( \lambda \) values from the equation

\[
\det(A - \lambda I) = 0
\]
Matlab provides a simple way to get eigenvalues and eigenvectors of a square matrix:

```matlab
>> A = [1 2; 3 4];
>> [evec, lambda] = eig(A)

evec =
   -0.8246   -0.4160
    0.5658   -0.9094

lambda =
   -0.3723       0
     0    5.3723
```
Schrödinger Equation

- Schrödinger Equation (SE) is a one of the good example for an eigenvalue problem.

- We will generally consider numerical solution to the one dimensional and time independent problems. Therefore, SE for this case is:

\[
\frac{-\hbar^2}{2m} \frac{d^2\Psi}{dx^2} + V(x)\Psi = E\Psi
\]

- For simplicity, we will also consider that \( \hbar = m = 1 \)

\[
\frac{1}{2} \frac{d^2\Psi}{dx^2} + V(x)\Psi = E\Psi
\]

- Here the problem is to solve energy \( E \) and WaveFunction \( \Psi \) at the same time!

- There are a few number of numerical solution method for the SE. We will use a finite difference matrix method.
A Particle in an Infinite Well

Analytical solution:

\[ E_n = \frac{\hbar^2 \pi^2 n^2}{8mL^2} \quad n = 1, 2, 3, \ldots \]

\[ \Psi_n = \frac{1}{\sqrt{L}} \cos\left(\frac{n\pi x}{2L}\right) \quad \text{if } n \text{ is odd} \]

\[ \Psi_n = \frac{1}{\sqrt{L}} \sin\left(\frac{n\pi x}{2L}\right) \quad \text{if } n \text{ is even} \]

Note that for \( \hbar = m = L = 1 \) we have

\[ E_n = \frac{\pi^2 n^2}{8} \approx 1.2337 n^2 \]

\[ \begin{array}{cc}
    n & E \\
    1 & 1.2337 \\
    2 & 4.9348 \\
    3 & 11.1033 \\
    4 & 19.7392 \\
    5 & 30.8425 \\
    6 & 44.4132 \\
    \ldots & \ldots \\
    \end{array} \]
Numerical solution using finite difference method:

\[-\frac{1}{2} \frac{d^2 \Psi}{dx^2} + V(x) \Psi = E \Psi\]

\[-\frac{1}{2} \frac{\Psi_{i+1} - 2\Psi_i + \Psi_{i-1}}{(\Delta x)^2} + V_i \Psi_i \approx E \Psi_i\]

or

\[\Psi_{i+1} = 2\Psi_i - \Psi_{i-1} - 2(\Delta x)^2 (E - V_i) \Psi_i\]

or

\[(\frac{1}{(\Delta x)^2} + V_i - E) \Psi_i - \frac{1}{2(\Delta x)^2} \Psi_{i-1} - \frac{1}{2(\Delta x)^2} \Psi_{i+1} = 0\]

\[\Delta x = \frac{2L}{5}\]

\[\Psi(-L) = \Psi_1 = 0\]

\[\Psi(+L) = \Psi_6 = 0\]
Matrix form:

\[
\begin{pmatrix}
\frac{1}{(\Delta x)^2} + V_2 - E & -\frac{1}{2(\Delta x)^2} & 0 & 0 \\
-\frac{1}{2(\Delta x)^2} & \frac{1}{(\Delta x)^2} + V_3 - E & -\frac{1}{2(\Delta x)^2} & 0 \\
0 & -\frac{1}{2(\Delta x)^2} & \frac{1}{(\Delta x)^2} + V_4 - E & -\frac{1}{2(\Delta x)^2} \\
0 & 0 & -\frac{1}{2(\Delta x)^2} & \frac{1}{(\Delta x)^2} + V_5 - E
\end{pmatrix}
\begin{pmatrix}
\Psi_2 \\
\Psi_3 \\
\Psi_4 \\
\Psi_5
\end{pmatrix}
= 
\begin{pmatrix}
0 \\
0 \\
0 \\
0
\end{pmatrix}
\]

For \( L = 1 \) \((dx = 2/5 = 0.4)\) and free particle \((V_i = 0)\)

\[
\begin{pmatrix}
6.250 - E & -3.125 & 0 & 0 \\
-3.125 & 6.250 - E & -3.125 & 0 \\
0 & -3.125 & 6.250 - E & -3.125 \\
0 & 0 & -3.125 & 6.250 - E
\end{pmatrix}
\begin{pmatrix}
\Psi_2 \\
\Psi_3 \\
\Psi_4 \\
\Psi_5
\end{pmatrix}
= 
\begin{pmatrix}
0 \\
0 \\
0 \\
0
\end{pmatrix}
\]
% infwell-minimal.m  (minimal form)
% Matlab solution for a free particle in an infinite well
% See also: http://www1.gantep.edu.tr/~bingul/ep475/src/

L = 1.0;  % well width
n = 5;    % number of parts
dx = 2*L/n;
A = zeros(n-1);
for i=1:n-1
    for j=1:n-1
        if i==j+1 | i==j-1
            A(i,j) = -0.5/dx^2;
        end
    end
    A(i,i) = 1.0/dx^2;
end
A  % print matrix
[wfn E] = eig(A);
x = -L+dx:dx:L-dx;
fprintf('*** Energy eigenvalues upto n = %d ***
',n);
for i=1:n-1
    fprintf('n=%2d  --> E=%15.4f
',i,E(i,i));
    plot(x,wfn(:,i));
pause(1);
end
Results for n = 5

```matlab
>> infwell;

A =

6.2500   -3.1250         0         0
-3.1250    6.2500   -3.1250         0
         0   -3.1250    6.2500   -3.1250
         0         0   -3.1250    6.2500

*** Energy eigenvalues upto n = 5 ***

n= 1 --> E= 1.1936
n= 2 --> E= 4.3186
n= 3 --> E= 8.1814
n= 4 --> E= 11.3064
```
Results for $n = 50$

```
>> infwell;
*** Energy eigenvalues upto $n = 50$ ***
n= 1 --> E= 1.2333
n= 2 --> E= 4.9283
n= 3 --> E= 11.0705
n= 4 --> E= 19.6355
n= 5 --> E= 30.5897
n= 6 --> E= 43.8897
n= 7 --> E= 59.4831
n= 8 --> E= 77.3083
n= 9 --> E= 97.2950
n=10 --> E= 119.3644
n=11 --> E= 143.4292
n=12 --> E= 169.3946
```
Results for n = 500

```matlab
>> infwell;
*** Energy eigenvalues upto n = 500 ***
n= 1 --> E= 1.2337
n= 2 --> E= 4.9347
n= 3 --> E= 11.1030
n= 4 --> E= 19.7382
n= 5 --> E= 30.8400
n= 6 --> E= 44.4080
n= 7 --> E= 60.4416
n= 8 --> E= 78.9402
n= 9 --> E= 99.9031
n=10 --> E= 123.3295
n=11 --> E= 149.2184
n=12 --> E= 177.5687
.
.
.
```
### Wave functions

- **n = 1**
- **n = 2**
- **n = 3**

### Eigenvalues

<table>
<thead>
<tr>
<th>n</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.2337</td>
</tr>
<tr>
<td>2</td>
<td>4.9348</td>
</tr>
<tr>
<td>3</td>
<td>11.1033</td>
</tr>
<tr>
<td>4</td>
<td>19.7392</td>
</tr>
<tr>
<td>5</td>
<td>30.8425</td>
</tr>
<tr>
<td>6</td>
<td>44.4132</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A Particle in a Finite Square Well

Depending on value of $V_0$ we can extend the limits of the SE.

For bound states ($E < V_0$) we can integrate the SE within the range (say) $[-5L, 5L]$. Here we may assume that the wave function vanishes at the boundaries.

We need a small modification for the `infwell.m`.

You can find the computer implementation `sqrwell.m` at:
http://www1.gantep.edu.tr/~bingul/ep475/src/
Results for $n = 500$

```plaintext
>> sqrwell;
*** Energy eigenvalues for bound states ***
n= 1 --> E= 1.0149
n= 2 --> E= 4.0516
n= 3 --> E= 9.0839
n= 4 --> E= 16.0596
n= 5 --> E= 24.8787
n= 6 --> E= 35.3246
n= 7 --> E= 46.6655
```
Wavefunctions

Infinite well

Finite well

\[ V_0 \]
Simple Harmonic Oscillator

Hamiltonian of the particle:

\[ \hat{H} = \frac{\hat{p}^2}{2m} + \frac{1}{2}m\omega^2\hat{x}^2 \]

Energy eigenvalues:

\[ E_n = \hbar\omega \left( n + \frac{1}{2} \right) \quad n = 0, 1, 2, \ldots \]

Wavefunctions

\[ \psi_n(x) = \frac{1}{\sqrt{2^n n! \left( \frac{m\omega}{\pi\hbar} \right)^{1/4}}} \cdot e^{-\frac{m\omega x^2}{2\hbar}} \cdot H_n \left( \sqrt{\frac{m\omega}{\hbar}} x \right) \]

\[ H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} \left( e^{-x^2} \right) \]
\[ E_n = \hbar \omega \left( n + \frac{1}{2} \right) \quad n = 0, 1, 2, \ldots \]

For \( \hbar = w = m = 1 \):

<table>
<thead>
<tr>
<th>( n )</th>
<th>( E )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>2</td>
<td>1.5</td>
</tr>
<tr>
<td>3</td>
<td>2.5</td>
</tr>
<tr>
<td>4</td>
<td>3.5</td>
</tr>
<tr>
<td>5</td>
<td>4.5</td>
</tr>
<tr>
<td>6</td>
<td>5.5</td>
</tr>
</tbody>
</table>

\[ \cdots \]
Results for n = 500

```matlab
>> harmonic;
*** Energy eigenvalues for first 10 states ***
\[
\begin{align*}
  n=1 & \quad \rightarrow \quad E= \quad 0.5000 \\
  n=2 & \quad \rightarrow \quad E= \quad 1.4998 \\
  n=3 & \quad \rightarrow \quad E= \quad 2.4995 \\
  n=4 & \quad \rightarrow \quad E= \quad 3.4990 \\
  n=5 & \quad \rightarrow \quad E= \quad 4.4983 \\
  n=6 & \quad \rightarrow \quad E= \quad 5.4975 \\
  n=7 & \quad \rightarrow \quad E= \quad 6.4966 \\
  n=8 & \quad \rightarrow \quad E= \quad 7.4954 \\
  n=9 & \quad \rightarrow \quad E= \quad 8.4941 \\
  n=10 & \quad \rightarrow \quad E= \quad 9.4927 \\
\end{align*}
\]```
Lennard-Jones Potential

The Lennard-Jones potential is a mathematically simple model that approximates the interaction between a pair of neutral atoms or molecules:

\[ V_{LJ} = 4\varepsilon \left[ \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^{6} \right] \]

Note that there is no analytical solution for the SE:

\[ -\frac{1}{2} \frac{d^2\Psi}{dr^2} + V_{LJ}\Psi = E\Psi \]
Sample run for $n = 600$ parts, $\sigma = 1$ and $\epsilon = 100$

>> LenardJones;
*** Energy eigenvalues for bound states ***
$n = 1 \rightarrow E = -66.2767$
$n = 2 \rightarrow E = -22.9936$
$n = 3 \rightarrow E = -4.1411$
Consider a 2D problem and assume that we break up space into a lattice of points with \( \Psi(i, j) \) being the value of the wavefunction at location \((i, j)\) on the lattice. SE \( \hbar = m = 1 \)

\[
- \frac{1}{2} \left[ \frac{d^2 \Psi}{dx^2} + \frac{d^2 \Psi}{dy^2} \right] + V(x, y)\Psi = E\Psi
\]

or

\[
- \frac{1}{2} \left[ \frac{\Psi(i+1, j) + \Psi(i-1, j) - 2\Psi(i, j)}{\Delta x^2} \right] + V(i, j)\Psi(i, j) = E\Psi(i, j)
\]

\[
- \frac{1}{2} \left[ \frac{\Psi(i, j+1) + \Psi(i, j-1) - 2\Psi(i, j)}{\Delta y^2} \right] + V(i, j)\Psi(i, j) = E\Psi(i, j)
\]
For \( dx = dy = h \), the last finite difference equation can be written in matrix form as:

\[
\begin{pmatrix}
\frac{1}{h^2} + V_{22} - E & -\frac{1}{2h^2} & 0 & \cdots \\
-\frac{1}{2h^2} & \frac{1}{h^2} + V_{23} - E & -\frac{1}{2h^2} & \cdots \\
0 & -\frac{1}{2h^2} & \frac{1}{h^2} + V_{24} - E & \cdots \\
\vdots & \vdots & \vdots & \ddots
\end{pmatrix}
\begin{pmatrix}
\Psi_{12} \\
\Psi_{13} \\
\vdots
\end{pmatrix}
= \begin{pmatrix}
0 \\
0 \\
\vdots
\end{pmatrix}
\]

Using the matrix method for finding eigenvalues and vectors is not an easy task and requires large amounts of computer time.

**e.g.** if there are 50 grid elements we have

* a 2500x2500 matrix to deal with in 2D \((50 \text{ MB RAM})\)

* a 125000x125000 matrix to deal with in 3D \((125 \text{ GB RAM})\)
Alternative Methods to Solve SE

Finite Difference Method for solving boundary value and eigenvalue problems are time and memory consuming.

There are alternative methods to solve Schrödinger Equation:

- **Shooting Method**
  good at symmetric potentials

- **Matching Method**
  good at all potentials & 1D problems

- **Variational Method**
  good at all potentials & dimensions but for ground state only

See web page for the example programs.
Shooting Method

- Schrödinger Equation (m=h-bar=1)

\[- \frac{1}{2} \frac{d^2}{dx^2} + V(x)\psi = E\psi\]

- This equation is similar to many other second order differential equations, with one interesting twist, the energy, E, is also unknown.

- The numerical solution of this equation must determine both \(\psi(x)\) and E.
- When the potential is symmetric, one can use the symmetry to an advantage. Namely, the wave functions can be written as purely odd or purely even functions of $x$.

- An even parity solution at $x=0$ requires

$$\psi(0) = 0 \quad \psi'(0) = 1$$

- An odd parity solution at $x=0$ requires

$$\psi(0) = 1 \quad \psi'(0) = 0$$
The method of solution for

\[ \frac{d^2 \psi}{dx^2} = 2(V(x) - E)\psi \]

is to pick a values of E, propagate the solutions (for a given parity) to large x to determine if they match the boundary condition at large x. If they do not, adjust the value of E and try again.
Infinite Well:

\[ \frac{d^2 \psi}{dx^2} = -2E \psi \]

\[ \psi(0) = 0 \quad \psi'(0) = 1 \]
% Shooting method for a particle in an infinite well
% Cromer-Euler solution

L = 1; % well width
n = 1000; % number of parts
dx = L/n;
parity = 1;
E = 1; % energy estimate
dE = 0.1;
Enew = E - dE;
psinew = 0;

while 1
    x=0; psi=1-parity; z=parity;
    for i=1:n
        z = z - 2*E*psi*dx;
        psi = psi + z*dx;
        x = x + dx;
    end
    if psi*psinew<0, dE = -dE/2; end
    E = E + dE;
    psinew = psi;
    if abs(E-Enew)<1.0e-6, break; end
    Enew = E;
end
fprintf('Parity = %d and Energy = %f\n',parity, E);
References:

[3]. Numerical Methods in Engineering with MATLAB,  
[4]. Essential MATLAB for Engineers and Scientist, 3rd Ed  
[5]. Computational Physics,  