Chapter 10

Quantum Mechanics

Time-independent Schroedinger Equation

$$-\frac{\hbar^2}{2m}\frac{d^2\phi(x)}{dx^2} + V(x)\,\phi(x) = E\,\phi(x) \qquad \textbf{Eq. (1)}$$

$$\hat{H}\,\phi(x) = E\,\phi(x)$$

$$\hat{H} = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x)$$

$\phi(x)$ is an *eigenstate* of the Hamiltonian operato

with the *eigenvalue* E

Time Independent Schroedinger Equation (cont.)

- In the time-independent Schroedinger Equation,

$$-\frac{\hbar^2}{2m}\frac{d^2\phi(x)}{dx^2} + V(x)\,\phi(x) = E\,\phi(x).$$

  the constant *E* is an unknown

- The solution has to obey boundary conditions fixed by physical consideration.

Ground state energy of an infinite square well

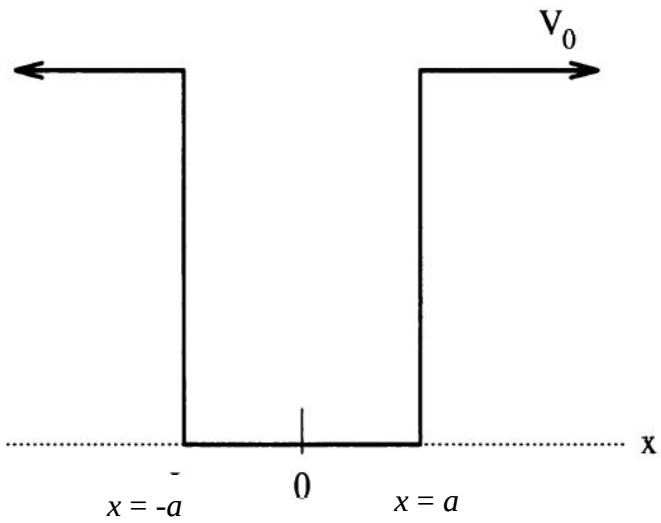


Figure 10.1: Potential energy for a particle in a box. The potential is $V = 0$ inside the box (i.e., for $|x| \leq a$ and V_0 outside. If $V_0 = \infty$, then the box is said to have "hard" walls, while if V_0 is finite, the walls are "soft."

Analytical eigen values

We first consider potentials for which a particle is confined to a specific region of space. Such a potential is known as the infinite square well and is described by

$$V(x) = \begin{cases} 0 & \text{for } |x| \le a \\ \infty & \text{for } |x| > a. \end{cases} \qquad (16.13)$$

For this potential, an acceptable solution must vanish at the boundaries of the well. We will find that the eigenstates $\phi_n(x)$ can satisfy these boundary conditions only for specific values of the energy E_n.

$$E_n = \frac{p_n^2}{2m}; p_n = \frac{h}{\lambda_n}; n\frac{\lambda_n}{2} = L = 2a$$

$$\Rightarrow \lambda_n = \frac{4a}{n}$$

$$\Rightarrow E_n = \frac{1}{2m}\left(\frac{h}{\lambda_n}\right)^2 = \frac{n^2\pi^2\hbar^2}{8ma^2} \qquad n = 1, 2, 3, \ldots$$

Analytical eigen solutions

The normalized eigenstates (eigen solutions) have the form

Show analytically that the energy eigenvalues of the infinite square well are given by $E_n = n^2\pi^2\hbar^2/8ma^2$, where n is a positive integer. Also show that the normalized eigenstates have the form

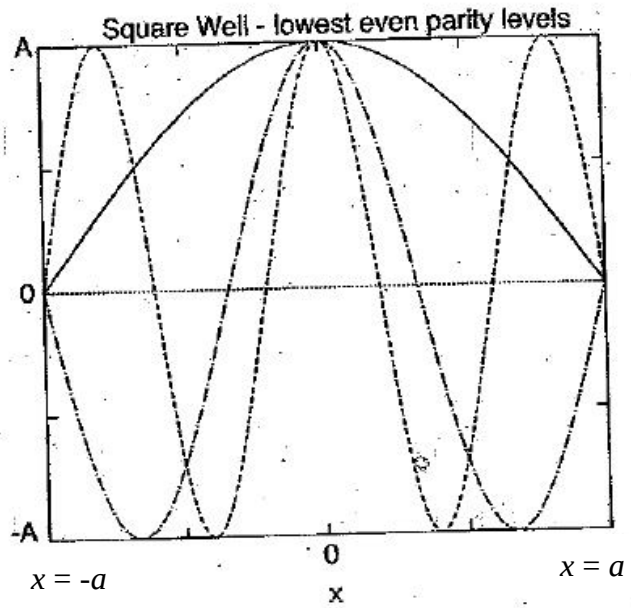
$$\phi_n(x) = \frac{1}{\sqrt{a}} \cos \frac{n\pi x}{2a}, \quad n = 1, 3, \ldots \quad \text{(even parity)} \qquad (16.14a)$$

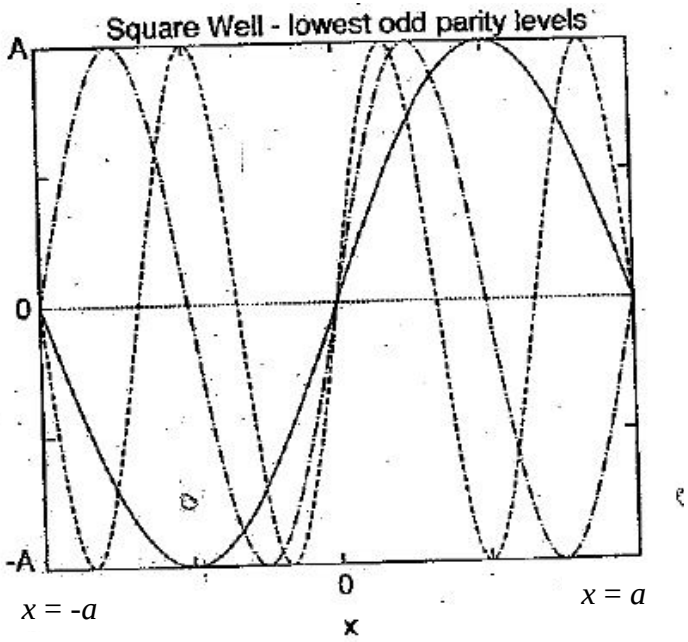
$$\phi_n(x) = \frac{1}{\sqrt{a}} \sin \frac{n\pi x}{2a}, \quad n = 2, 4, \ldots \quad \text{(odd parity).} \qquad (16.14b)$$

Normalisation condition $\int_{-\infty}^{\infty} \phi_n(x)\phi_n^*(x)\,dx = 1$

allows us to obtain the normalisation constant of the eigen state, $\phi_n(x)$, i.e., $\frac{1}{\sqrt{a}}$

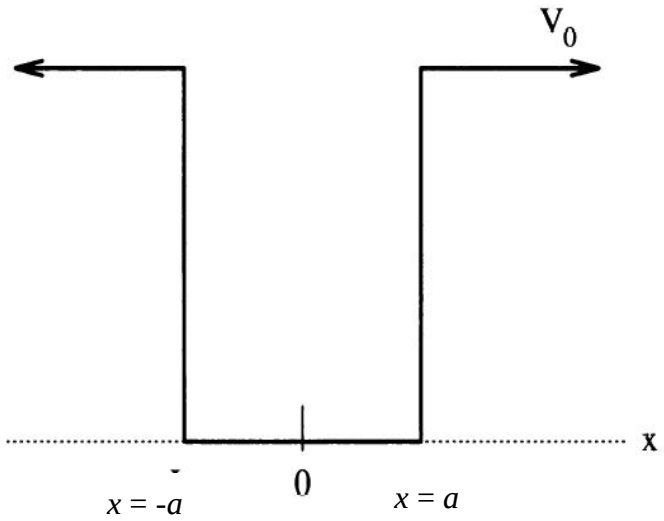
Even and Odd eigen states





Numerical value of V_0

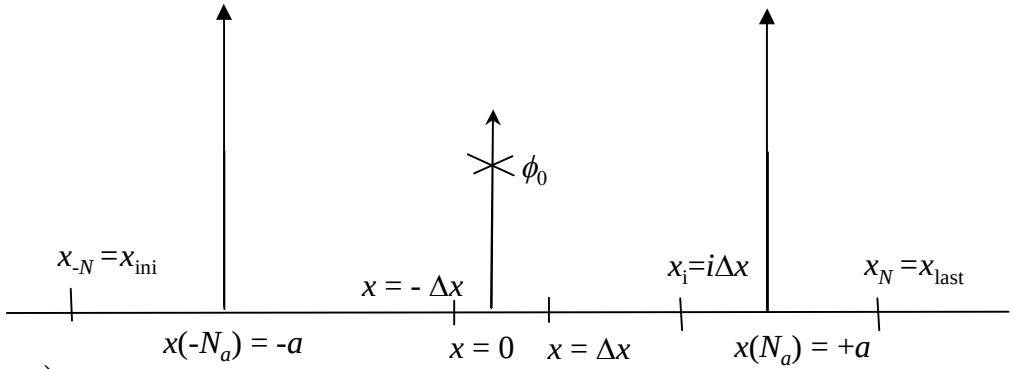
- The potential takes on the form of $V_i = \begin{cases} V_0, & |i| \le N_a \\ 0, & |i| > N_a \end{cases}$



- When solving infinite square well numerically, we can't set V_0 to ∞. To mimic "infinity" numerically, simply set V_0 to a very large value, i.e., 1000.

Discretisation

- Choose unit: $\hbar$, m = 1.



We let: $V_i \equiv V(x_i)$, $\phi_i \equiv \phi(x_i) = \phi(i\Delta x)$

$x_i = i\Delta x$; $x_{-N} = x_{ini}$; $x_N = x_{last}$; $\Delta x = (x_{last} - x_{ini})/N$, N number of spatial interval Δx.

i=-N,-N+1,…,0, 1,2,…, N-1, N indexes a grid point x_i.

The size of the well is 2a; its edges are located at $x(-N_a) = -a$ and $x(N_a) = a$.

We need to specify (as an input) x_{ini}, x_{last}; a, N.

Δx, N_a are calculated based on these.

Discretising the TISE

- Discretise the TISE on the grid as defined earlier:

$$\phi_i = \phi(i\Delta x)$$

$$\frac{d^2\phi_i}{dx^2} \simeq \frac{\phi_{i+1} + \phi_{i-1} - 2\phi_i}{(\Delta x)^2}$$

$$-\frac{\hbar^2}{2m}\frac{d^2\phi_i}{dx^2} \simeq -\frac{\hbar^2}{2m}\frac{\phi_{i+1} + \phi_{i-1} - 2\phi_i}{(\Delta x)^2} \simeq (E - V_i)\phi_i$$

$$\phi_{i+1} = 2\phi_i - \phi_{i-1} - 2(\Delta x)^2 (E - V_i)\phi_i$$

Initial conditions

- To begin with the first step, we need to supply *two* initial values at $\phi(x=0)\equiv\phi_0, \phi'(x=0)$
- Generically, two initial values are required for solving a second order differentiation. Choosing an initial value at $x=0$ for $\phi'(x=0)\equiv\phi_0'$ is equivalent to choosing an initial value at ϕ_{-1}, since $\phi'(0)\approx\frac{\phi(0)-\phi(-\Delta x)}{\Delta x}\Rightarrow\phi_{-1}=\phi_0-\Delta x\phi_0'$

Symmetry of the solution

- For the case of the infinite potential well, the potential is symmetric about $x=0$. We hence expect the solutions correspond to this potential, i.e., ϕ, should also contain a corresponding symmetry.
- The solutions are expected to display either positive or negative parity symmetry, ie. $\phi(-x)=\phi(x)$ +ve parity; $\phi(-x)=-\phi(x)$ -ve parity.
- Parity will be used to prepare the initial conditions for our numerical solution scheme.

Initial conditions

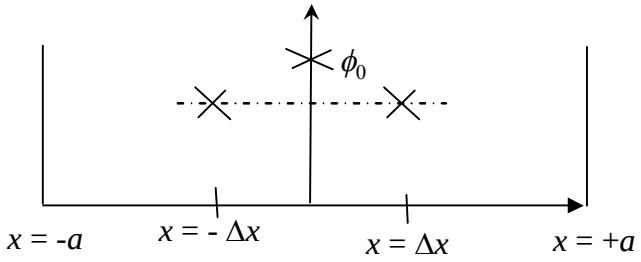
- To begin with the first step, we need to supply *two* initial values at $\phi(x=0)\equiv\phi_0, \phi'(x=0)\equiv\phi_0'$
- Generically, two initial values are required for solving a second order differentiation. Choosing an initial value at $x=0$, is equivalent to choosing an initial value at ϕ_{-1}, since $\phi'(0)\approx\frac{\phi(0)-\phi(-\Delta x)}{\Delta x}\Rightarrow\phi_{-1}=\phi_0-\Delta x\phi_0'$

Initial conditions for ϕ_0, ϕ_{-1}

- The solutions can be distinctively be categorised into either positive or negative parity ones.
- For +ve parity case: $\phi(\Delta x) = \phi(-\Delta x)$

$$\phi'_0 = \left.\frac{d\phi}{dx}\right|_{x=0} \approx \frac{\phi(0) - \phi(-\Delta x)}{\Delta x} = 0$$

- Choose $\phi_{-1} = \phi_0 = 1$

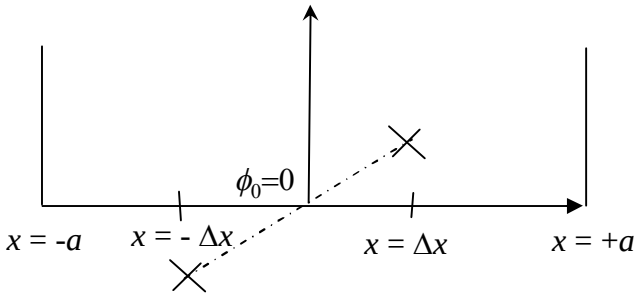


Initial conditions for ϕ_0, ϕ_{-1}

- For –ve parity case: $\phi(\Delta x) = -\phi(-\Delta x)$

$$\phi'_0 = \left.\frac{d\phi}{dx}\right|_{x=0} \approx \frac{\phi(\Delta x) - \phi(-\Delta x)}{2\Delta x} = \frac{2\phi(\Delta x)}{2\Delta x} = \frac{\phi(\Delta x)}{\Delta x}$$

- Choose $\phi_0 = 0$ $\left.\frac{d\phi}{dx}\right|_{x=0} = 1 \Rightarrow \phi(-\Delta x) = \phi_{-1} = -\Delta x$



Boundary condition

- But now, there is the boundary condition to take care of: we need the solution to be such that

$$\phi(x=-a)=\phi(x=a)=0$$

- This is a physical requirement as we know that there is no particle to be found outside the potential well, according to the probabilistic interpretation of $\phi\phi^*$.
- But before we arrive at the end points we would not know whether such boundary condition is obeyed. We hence have to adjust the so-far unknown values of *E*, the eigen value of the TISE, to achieve this requirement.

Expected eigen values

- We will find that, if the size of the box is chosen to be $L = 2a = 2$, mass $m = 1$, then when E is chosen to be exactly 1.2337 will the boundary condition get statisfied.
- The next lowest eigen value for which the boundary condition get satisfied is 4×1.2337.
- In fact only for $E = 1.2337n^2$, $n = 1, 2, 3, \ldots$ will the boundary condition be met.
- For values of E other than $1.2337n^2$, the solution divert to $\pm\infty$ at the boundaries. The discrete values of eigenvalues corresponds to the quantise energy levels of the infinite quantum square well system.
- Note: We have chosen the unit where $\hbar = 1$ so that $E_n = \frac{n^2\pi^2\hbar^2}{8ma^2} = 1.2337n^2$

Shooting method

- To this end, we use what is called the SHOOTING method: we iterate Eq. (1) with a dynamically adjusted value of *E* so that the code could find for itself automatically a value of *E* such that the boundary condition is met.
- Hint: choose a value of *E* close to 1.2 as an initial guess to *E*.

ever-decreasing

EXAMPLE 10.1 Interactive shooting method routine `calculate`

- Input are the number N and the size Δx of the spatial grids, the initial guess for the energy E and increment ΔE, and the cutoff parameter b (see below).
- Implement $\psi'(0) = 0$ for an even parity solution by setting $\psi_0 = \psi_{-1} = 0$.
- Initialize variable *last_diverge* (which keeps track of the direction of the diverging trend) to zero (since we do not know this direction *à priori*). If the subsequent calculations to evaluate ψ for a given value of E exits by detecting a diverging trend, the sign of the last value of ψ evaluated indicates the direction.
- Main loop in which we iterate and home-in to a valid eigenvalue E:
  - ▷ For the current value of E, loop through the index $i = 0, 1, ..., N-1$ to successively compute the wave function $\{\psi_i\}$.
    - Calculate ψ_{i+1} from ψ_i and ψ_{i-1} using $\psi_{i+1} = 2\psi_i - \psi_{i-1} - 2(E - V(x_i))(\Delta x)^2\psi_i$.
    - Check if $|\psi_{i+1}| > b$ for a suitable cutoff value b.
      - ▷ If yes, assume that ψ is diverging. Exit the loop over i and display the computed ψ.
      - ▷ Otherwise, continue the loop with the next value i.
  - ▷ Display the computed ψ and current value of ΔE, and wait for key input to see if we should continue further.
    - If ΔE is small enough, we may conclude that the current E is acceptable (if the displayed ψ looks good), or reject this calculation (if it is not). Exit the subroutine.
    - If further iteration is indicated, continue.
  - ▷ If the last value of ψ evaluated ($\psi(i+1)$) and *last_diverge* are of different signs, turn around the direction of varying E and halve its magnitude by assigning $\Delta E = -\Delta E/2$.
  - ▷ Update the trial value of E to $E + \Delta E$.
  - ▷ Replace *last_diverge* by the sign of $\psi(i+1)$.
  - ▷ Go back to the beginning of the homing-in process and iterate.

Descrised eigen energies

- For $L = 2a = 2$, when E is chosen exactly as 1.2337, the boundary condition get statisfied.
- The next lowest eigen value for which the boundary condition get satisfied is 4×1.2337.
- In fact only for $E = 1.2337n^2$, $n = 1, 2, 3, \ldots$ will the boundary condition be met.
- For values of E other than $1.2337n^2$, the solution divert to $\pm\infty$ at the boundaries.
- The discrete values of eigenvalues corresponds to the quantise energy levels of the infinite quantum square well system.
- Note that the agreement between numerical and analytical eigen value can only be achieved if the discretisation is made sufficiently fine ($N \sim 10^5$).

Normalisation

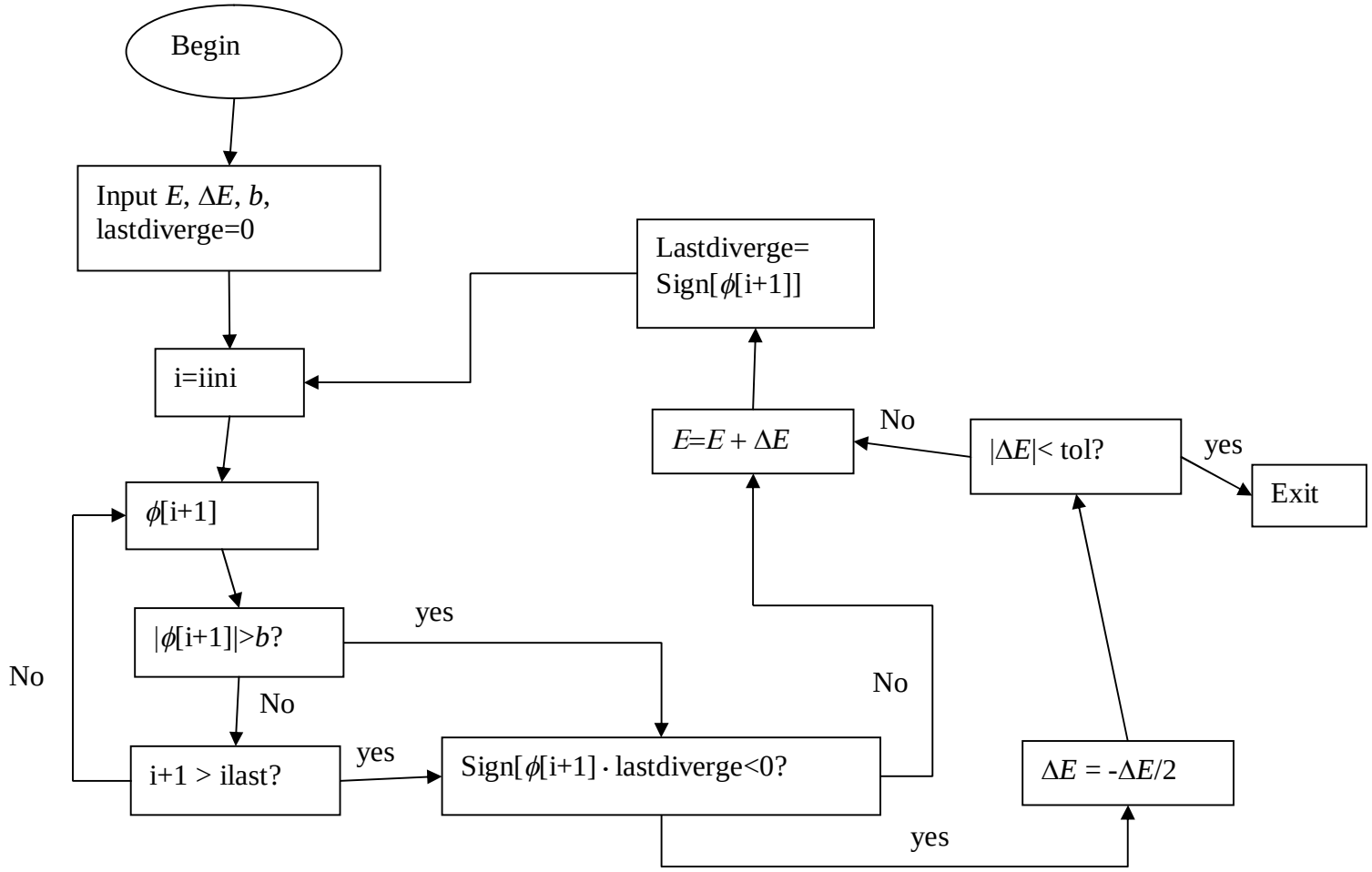
- Once the eigen solution is obtained, with the right eigen values *E* is found with the shooting method, we will have to rescale the solution according to the normalisation condition:

$$A = \int_{-a}^{a} \phi^{*}(x)\phi(x)\,dx$$

$$\phi_i(x) \to \phi_i^N(x) = \phi_i(x)/\sqrt{A} \text{ so that } \int_{-a}^{a} \phi_i^{N*}(x)\phi_i^N(x)\,dx = \int_{-a}^{a} \frac{\phi_i^{*}(x)}{\sqrt{A}}\frac{\phi_i(x)}{\sqrt{A}}\,dx = 1$$

- In Mathematica, this can be done via the command:
- Sumphi=Sum[phi[i]^2*Deltax,{i,iini,ilast}];
- phi[i]=phi[i]/Sqrt[Sumphi] for all i

Flow chart of shooting method

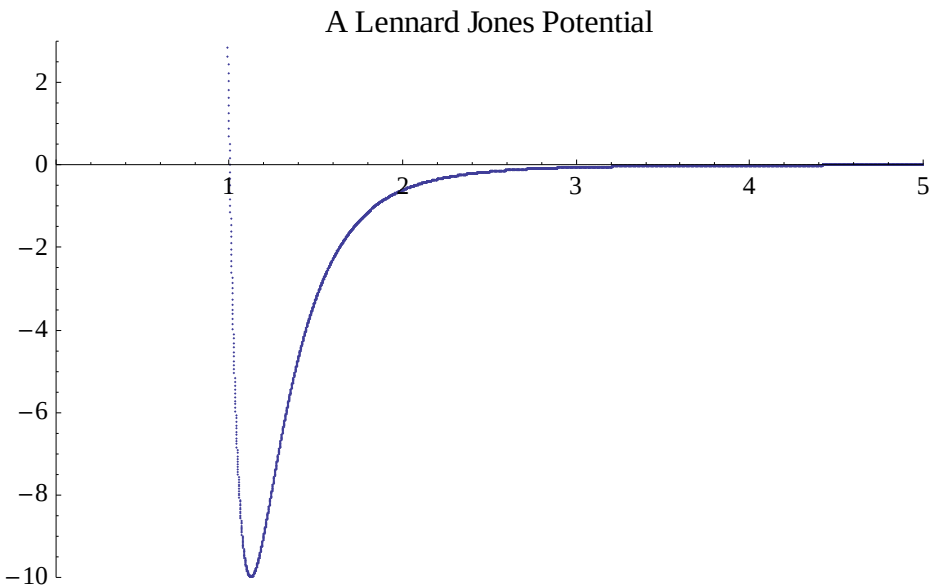


Sample code 10.1

- Sample code 10.1 implements the shoot method to solve the 1D infinite square well. If $V0$ is set effectively very large at the edge of the well, i.e., $V(-a)=V(a)=V0$, it mimics an infinite 1-D square well.

1D non-symmetrical potential: LJ Potential

- Consider a general potential where the solutions have not any parity symmetry, e.g., a Lennard-Jones potential,



$$V(x) = 4\varepsilon\left[\left(\frac{\sigma}{x}\right)^{12} - \left(\frac{\sigma}{x}\right)^{6}\right]$$

$$\varepsilon = 10; \sigma = 1$$

Initial conditions for ϕ_L

- Iterate $\phi_{i+1} = 2\phi_i - \phi_{i-1} - 2(\Delta x)^2 (E - V_i)\phi_i$
- from the left region from

  x_L, where $\phi(x_L) = 0, \phi'(x_L) = \delta \to 0,$ with $\delta > 0$
- Call this ϕ_L
- To this end, we need two initial values:

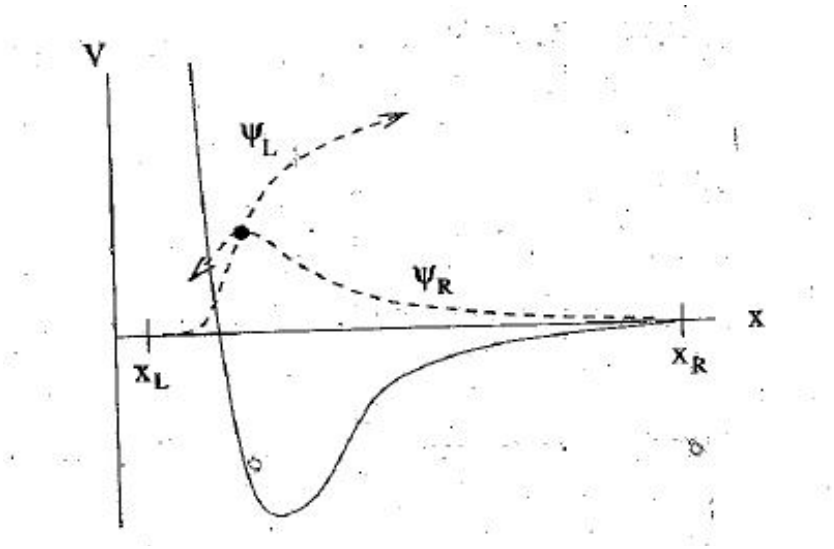
  $\phi(x_L), \phi(x_L - \Delta x)$
- Since $\phi'(x_L) = \delta \approx \frac{\phi(x_L) - \phi(x_L - \Delta x)}{\Delta x}$

  $\to \phi(x_L - \Delta x) = \phi(x_L) - \delta \Delta x$

Initial conditions for ϕ_R

- Repeat the procedure from the right region from x_R, where $\phi(x_R) = 0, \phi'(x_R) = \delta \to 0$
- Call this $\phi(x_R)$

$$\phi(x_R - \Delta x) = \phi(x_R) - \delta \Delta x$$



Matching procedure

- Identify the point x where V is minimal, call it $x_{\min}$.
- Scale both or either of ϕ_L, ϕ_R so that $\phi_L(x_{\min}) = \phi_R(x_{\min})$
- Vary E until $\phi'_L(x_{\min}) = \phi'_R(x_{\min})$

Assign *match*= $\text{Sign}[\phi'_L(x_{\min}) - \phi'_R(x_{\min})]$

If *match* different from pervious trail, $\Delta E = -\Delta E/2$
$E = E + \Delta E$
Iterate the loop again with the new E.

- Run the LJ potential with $\varepsilon = 10$, $\sigma = 1$, $\Delta x = 0.01, x_R = 5.0$, $x_L = 0.5$.

Sample code 10.2

- Sample code 10.2 is modified from sample code 10.1 by replacing the potential to the LJ form.
- To make things happen one in a step, sample code implements the iteration for phiL from the left hand side and independently, phiR from the right side.
- No matching procedure is carried out. Once you know how to iterate from both sides, then proceed to sample code 10.3 where the full matching procedure is implemented.

Sample code 10.3

- Sample code 10.3 implement the full matching procedure for the LJ potential.
- We find slightly different answer as compared to that given by Giordano's 2nd edition, Figure 10.8, page 317.
- For the parameter values we find the first eigen value at E=-1.8899, and is insensitive to the 'meeting point' x_{Min}, be it 1.12246, obtained by minimising $V(x)$, or 1.4 as given by Giodarno.

Some details

- However the exact location of the left side point x_L is important: it can't be too close to $x=0$, the singularity of $V(x)$.
- We find a value of $x_L=0.65$ can do the job well.
- Any value closer to 0 renders the solution unstable, resulting in phiL(xR) and phiR(xL) diverging at the edges.

Some details

- We can monitor phiL(x_R) and phiR(x_L) for the numerical behaviour of the code.
- If these numbers diverge, then adjust the input parameters such as x_R, x_L, or to the lesser extent x_{Min}.
- A diverging values (i.e., very large values, e.g., phiR(xL) = 5534543342 3) signifies numerical error.
- Visually we can also minitor the graphs of phiL, phiR. They will not overlap with each other in the case of this numerical error mentioned.

Excited states with matching method

- Sample code 10.4, which is written in a separated package under the folder "TISE" calls the subroutine as implemented in sample code 10.3 (with some minor modification) to evaluate the excited state eigenenergies, E_1, E_2, E_3, …, E_{nlast}, for any potential function defined explicitly in the file "potential.m". Sample code 10.4 fine-tune the matching method based on sample code 10.3 to handle generic TISE.

Excited states with matching method

- Higher excitation states, i.e., other eigen values of the energy, can be obtain by systematically increase the input trial energy E.
- Say once you have obtained the lowest eigenenergy (a.k.a eigen value, energy), say $E=E_0$ = -1.8899 (which corresponds to the ground state energy), rerun the code with a new energy input of $E_1= E_0 + k|E_0|$, where k a small positive fractional number, e.g., k = 0.5.
- Try to adjust the value of k incrementally until the resultant eigenvalue from the code to be different from E_0.

Some details

- To this end, we introduce a cutoff b to control the behaviour of the solution.
- Should |phiR(xL)| or |phiL(xR)| larger than b, typically set to a large e.g. 50 or 1000, we readjust the position of xL so that it is slightly further away from the original one.
- It is found that for some cases of potential such as for LJ, an initial position of xL too “deep” into the region where V(x) is large, numerical convergence may not be achieved.

Some details

- In general, given a generic potential, trial and error has to be carried out to find out what are the optimum values for the following parameters:
- Initial guess of the energy, Eini;
- initial position of the xL (xini), xR (xlast),
- cut-off b,
- and k, which determine how the initial guess for the next excited state energy should be, as occurs in En=eigenEnergy[n-1]+k*Abs[eigenEnergy[n-1]].
- Fine tuning is a must in order to assure that the eigen energies obtained are authentic.
- Check against known numerical results or inspect the numerical solution of phiL, phiR to determine if the numerical results are reliable.

Sample code 10.4

- Sample code 10.4 is an improved code using matching method for generic potential.
- The potential form is defined in the file potential.m, and read in by Main.nb.
- In the sample code 10.4, four types of potentials defined.
- p = 1 for square potential barrier
- p = 2 for square LJ
- p = 3 for infinite well
- p = 4 for 1-D harmonic oscillator
- However the code only takes care of the symmetric case and does not provide asymmetric solutions.

1-D Variation Monte Carlo

VMC for 1-D TISE

Variation Monte Carlo method is an alternative method to solve the TISE, using stochastic approach.

Its strength: much efficient than the difference method when $d > 4$, where d is the dimension of the problem.

Discretisation

Discretise the 1-D TISE into N bins, each with a width of Δx :

$$H\phi(x) = \left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial^2 x} + V(x) \right]\phi(x) = -\frac{\hbar^2}{2m}\frac{\partial}{\partial x}\left(\frac{\partial \phi(x)}{\partial x} \right) + V(x)\phi(x);$$

$$x_{ini} \le x \le x_{ilast}; \Delta x = \frac{x_{ilast} - x_{iini}}{N};$$

$$\phi(x_i) \equiv \phi_i; x_i = x_{ini} + i\Delta x; i = 0, 1, 2, \ldots, ilast = IntegerPart[(xlast - xini)/\Delta x];$$

Central differencing

Derive $\frac{\partial^2 \phi(x_i)}{\partial^2 x_i}$ using central differencing:

$$\frac{\partial \phi_i}{\partial x_i} = \left(\frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} \right);$$

$$\frac{\partial^2 \phi(x_i)}{\partial^2 x_i} = \frac{\partial}{\partial x_i}\left(\frac{\partial \phi(x_i)}{\partial x_i} \right) \simeq \frac{\partial}{\partial x_i}\left(\frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} \right) \simeq \frac{1}{2\Delta x}\left(\frac{\partial}{\partial x_i}\phi_{i+1} - \frac{\partial}{\partial x_i}\phi_{i-1} \right)$$

$$= \frac{1}{2\Delta x}\left(\frac{\phi_{i+2} - \phi_i}{2\Delta x} - \frac{\phi_i - \phi_{i-2}}{2\Delta x} \right) = \frac{1}{4\Delta x^2}\left(\phi_{i+2} - 2\phi_i + \phi_{i-2} \right)$$

Discretisation (cont.)

$$H\phi_i = \left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial^2 x} + V_i \right]\phi_i \simeq -\frac{\hbar^2}{8m}\frac{1}{\Delta x^2}\left(\phi_{i+2} - 2\phi_i + \phi_{i-2}\right) + V_i\phi_i$$

$$\int \phi^*(x) H\phi(x)\,dx \simeq \sum_{i=iini}^{ilast} \phi_i^* H\phi_i \Delta x_i \simeq \sum_{i=iini}^{ilast} \phi_i^* \left[-\frac{\hbar^2}{8m}\frac{1}{\Delta x^2}\left(\phi_{i+2} - 2\phi_i + \phi_{i-2}\right) + V_i\phi \right]\Delta x_i =$$

$$\sum_{i=0}^{ilast} -\frac{\hbar^2}{8m}\frac{\phi_i^*}{\Delta x}\left(\phi_{i+2} - 2\phi_{i+1} + \phi_i\right) + V_i\phi_i^2\Delta x$$

Alternative differencing scheme

Alternatively, we could also arrive at the following expression if followed a different difference scheme:

$$\int \phi^*(x) H \phi(x) dx \simeq \sum_{i=0}^{ilast} -\frac{\hbar^2}{2m} \frac{\phi_i^*}{\Delta x} (\phi_{i+1} - 2\phi_i + \phi_{i-1}) + V_i \phi_i^2 \Delta x$$

The VMC algorithm

In VMC, the following procedure is followed:

- Discretise the simulation space into N bins of size Δx, each indexed by i = [iini,ilast].
- Provide a trial guess for ϕ, call it ϕ^0, conveniently chosen to be a constant, e.g., 1 in the simulation space, i.e., ϕ^0_i=1 for i=[iini,ilast].
- Evaluate $$E^* = \frac{\int \phi^*(x) H \phi(x) dx}{\int \phi^*(x) \phi(x) dx} \simeq \frac{\sum_{i=iini}^{ilast} -\frac{\hbar^2}{2m}\frac{\phi_i^*}{\Delta x}(\phi_{i+1} - 2\phi_i + \phi_{i-1}) + V_i \phi_i^2 \Delta x}{\sum_{i=iini}^{ilast} \phi_i^* \phi_i \Delta x}$$

The VMC algorithm (cont.)

- While n ⩽ nlast, iterate:
  - Randomly choose a bin, i=iran, shift $\phi_{iran} \rightarrow \phi_{iran}^{trial} = \phi_{iran} + \delta\phi_{iran}$, where $\delta\phi_{iran}$ = Random[Real] = [0,1].
  - Evaluate $E_{trial}^{*} \simeq \dfrac{\sum_{i=iini}^{ilast} -\dfrac{\hbar^2}{8m}\dfrac{\phi_i^{trial*}}{\Delta x}\left(\phi_{i+2}^{trial} - 2\phi_i^{trial} + \phi_{i-2}^{trial}\right) + V_i\left(\phi_i^{trial}\right)^2 \Delta x}{\sum_{i=iini}^{ilast} \phi_i^{trial*}\phi_i^{trial}\Delta x}$
  - If $E_{trial}^{*} < E^{*}$, accept the shift, i.e, $\phi_{iran} = \phi_{iran}^{trial}$, and let $E^{*} = E_{trial}^{*}$, else reject the move.
- Upon exiting the loop after nlast step, normalised the function ϕ:

$$A = \sum_{i=iini}^{ilast} \phi_i^{*}\phi_i \Delta x\,, \quad \phi_i \rightarrow \frac{\phi_i}{\sqrt{A}}$$

Sample code 10.5 (VMC.nb)

- Sample code 10.5 implements the VMC algorithm which reproduces the expected ground state energy of the LJ potential quite satisfactorily. The ground state is known to be lying at E_0 = -1.8899. The sample code 10.5 gives E_0 = -1.91479 (with some error), running for 50000 steps, with N=70 bins.

- The code also calculates the GS energy of harmonic oscillator with potential $V = \frac{1}{2} kx^2$. With the mass m=1, and the spring constant k = 1, the expected GS of the harmonic oscillator is 0.5, which is also reproduced by sample code 10.5.

- Comment: This is a non-optimised method, and could be further improved using e.g., simulated annealing, or importance sampling method.

- In principle, the longer the code is run (i.e., if you wait longer), and with larger number of bins, the code converges to produce better quality results.