

# Numerical solution of the time-independent Schrödinger equation using Numerov's method

Joar Svartholm - josv0150(josv0150@student.umu.se)

March 9, 2018

## Contents

<b>1</b>	<b>Time independent Schrödinger equation</b>	<b>2</b>
<b>2</b>	<b>The harmonic oscillator</b>	<b>2</b>
<b>3</b>	<b>The Morse potential</b>	<b>5</b>
<b>4</b>	<b>End words</b>	<b>6</b>

# 1 Time independent Schrödinger equation

The time independent Schrödinger equation(TISE) is defined as

$$\left[ -\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + \mathcal{V}(x) \right] \Psi(x) = E\Psi(x) \iff \frac{d^2}{dx^2} \Psi(x) + \frac{2m}{\hbar^2} [E - \mathcal{V}(x)] \Psi(x) = 0. \quad (1)$$

Thus, by defining

$$k_j^2 = \frac{2m}{\hbar^2} (E - \mathcal{V}(x_j)) \quad (2a)$$

$$Y_j = \left( 1 + \frac{mh^2}{6\hbar^2} (E - \mathcal{V}(x_j)) \right) \Psi(x_j) \quad (2b)$$

in Eq. (1), Numerov's iteration scheme will be

$$Y_{j+1} = Y_j \left( 2 - \frac{2mh^2}{\hbar^2} \frac{(E - \mathcal{V}(x_j))}{1 + \frac{mh^2}{6\hbar^2} (E - \mathcal{V}(x_j))} \right) - Y_{j-1} \quad (3)$$

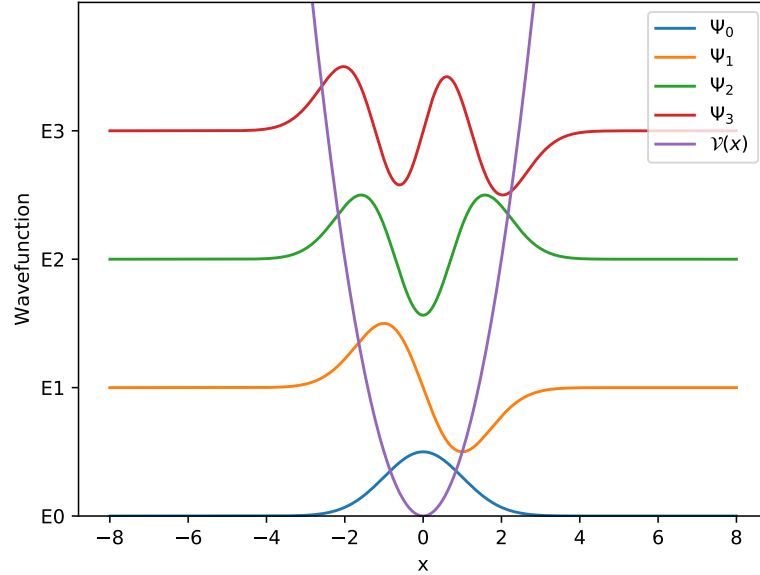
# 2 The harmonic oscillator

Consider now the potential of an harmonic oscillator

$$\mathcal{V}(x) = \frac{m\omega^2}{2} x^2. \quad (4)$$

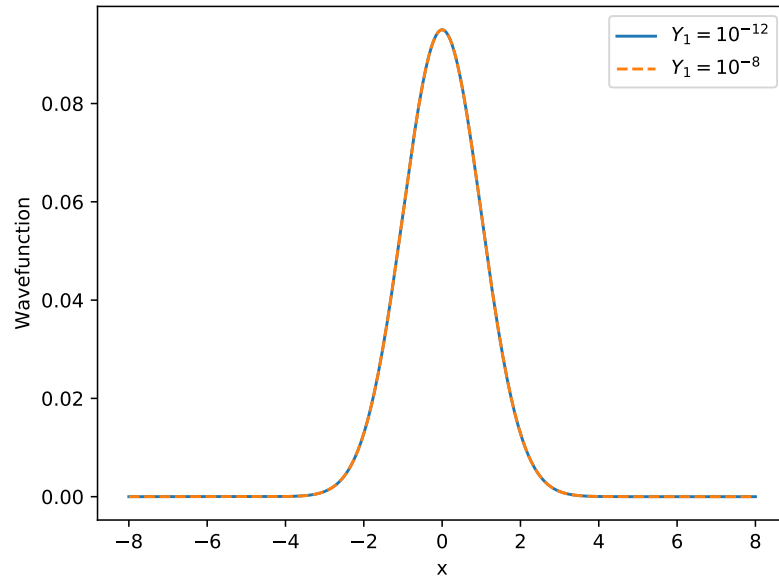
We will now find the wave function for the four first eigenstates using simulation units  $\hbar = m = \omega = 1$ . That is, for  $E = \{0.5, 1.5, 2.5, 3.5\}$ . Doing so using  $Y_0 = 0$  and  $Y_1 = 10^{-12}$  and normalizing the wavefunctions yields Fig. 1. As one can see in the figure the number of nodes of the wavefunction increases linearly with the state number. In the simulation, it does not matter what initial condition is used for  $Y_1$ . This can be shown by changing this to  $Y_1 = 10^{-8}$  and recomputing the wavefunction for the ground state. Since the wavefunctions are normalized the two solutions will be identical. This is shown in Fig. 2.

When doing these computations, Numerov's method is used from the left and from the right and the solution is matched in the middle. If it was only applied from the left the solution to the first eigenstate would blow up near the right boundary. This is shown in Fig. 3. In order to figure out why this is happening we must look at the analytic solution

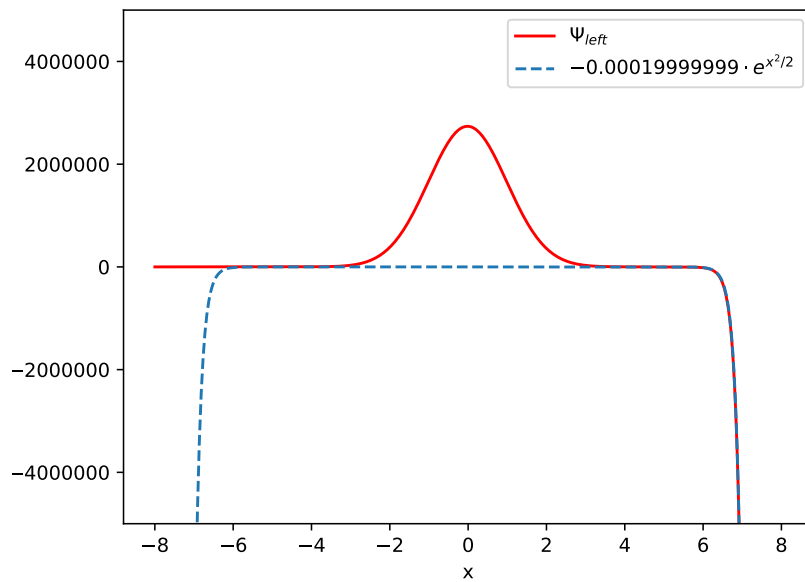


**Figure 1:** *The first four eigenstates of the harmonic oscillator*

of TISE. In fact, there are two solutions,  $\Psi_{01} = C_1 e^{-\alpha x^2/2}$  and  $\Psi_{02} = C_2 e^{\alpha x^2/2}$ , but for physical reasons  $\Psi_{02}$  is neglected since it cannot be normalized. When simulating, we probably get a solution that is the superposition of the two, but where  $C_2$  is really small and unnoticeable for small  $x$ . Thus for large  $x$  when simulation from one direction only, this solution will take over. As one can see in Fig. 3 an exponential of this form fits almost perfectly to the right side of the numerical solution using this method. This explains the odd behavior of the numerical solution.



**Figure 2:** Two solutions of the wavefunction for the ground state using different initial condition  $Y_1$ .



**Figure 3:** Ground state wavefunction when only computing from the left.

### 3 The Morse potential

Let us now consider the Morse potential

$$\mathcal{V}(R) = E_B \left[ 1 - e^{-a(R-R_e)} \right]^2 \quad (5)$$

where  $R$  is the distance between two atoms,  $R_e$  is the equilibrium distance,  $E_B$  is the binding energy and  $a$  is a scaling constant for the bond stiffness. In the center of mass frame, it is better to use spherical coordinates for the wave function,  $\Psi(\mathbf{R}) = \Psi(R)Y(\theta, \phi)$ . Taking the first orbital  $Y = Y_0^0 = 1/\sqrt{4\pi}$ , the problem reduces to a one dimensional problem since the orbital derivatives in the spherical laplacian derives to zero. Using the substitution  $\psi(R) = R\Psi(R)$ , TISE will reduce to

$$\left[ -\frac{\hbar^2}{2\mu} \frac{d^2}{dR^2} + \mathcal{V}(R) \right] \psi(R) = E\psi(R), \quad (6)$$

where  $\mu$  is the reduced mass and  $\mathcal{V}(R)$  is the potential in Eq. (5).

Since the energy levels are not known in opposite to the harmonic potential a good guess for the energy levels are needed in order to find the correct eigen energies and its corresponding wavefunction. In order to do that we make a Taylor expansion of Eq. (5) around the equilibrium distance  $R_e$

$$\mathcal{V}(R) = \mathcal{V}(R_e) + (R - R_e) \left( \frac{d\mathcal{V}}{dR} \right)_{R_e} + \frac{(R - R_e)^2}{2} \left( \frac{d^2\mathcal{V}}{dR^2} \right)_{R_e} + \mathcal{O}((R - R_e)^3). \quad (7)$$

At equilibrium, both the function and its derivative is zero, so only the third term is nonzero and thus, by neglecting higher order terms the potential can be approximated as

$$\begin{aligned} \mathcal{V}(R) &\approx \frac{(R - R_e)^2}{2} E_B \frac{d^2}{dx^2} [1 - e^{-ax}]^2 \Big|_{x=0} \\ &= \frac{(R - R_e)^2}{2} E_B 2 \frac{d}{dx} [1 - e^{-ax}] a e^{-ax} \Big|_{x=0} \\ &= a(R - R_e)^2 E_B [-a e^{-ax} + 2a e^{-ax}]_{x=0} \\ &= E_B a^2 (R - R_e)^2 \end{aligned} \quad (8)$$

when comparing this to the harmonic potential Eq. (4) it is clear that

$$\frac{\mu\omega^2}{2} = E_B a^2$$

must hold and the harmonic frequency can thus be written as

$$\omega = a\sqrt{\frac{2E_B}{\mu}} \quad (9)$$

For the harmonic oscillator, we know that the eigen energies are

$$E_n = \hbar\omega \left( n + \frac{1}{2} \right).$$

For the anharmonic oscillator we are considering, we can approximate the eigen energies using Eq. (9) and replacing the mass with the reduced mass to obtain

$$\tilde{E}_n = \hbar a \sqrt{\frac{2E_B}{\mu}} \left( n + \frac{1}{2} \right). \quad (10)$$

The ground state energy and the spacing is thus

$$\begin{aligned} \tilde{E}_0 &= \hbar a \sqrt{\frac{E_B}{2\mu}} \\ \Delta\tilde{E} &= \hbar a \sqrt{\frac{2E_B}{\mu}} \end{aligned}$$

## 4 End words

All C code is found in the code folder;  
`/home/josv0150/Documents/numPhys/numphys-numerov/code` including the code for plotting. If something needs to be recompiled simply type `make`. The plot functions are written in python3 and the line `python3 plotNumerov.py` or equivalent should do the trick. I have not tested to run these plotfunctions on sesam since I uploaded the codes remotely and I have not tried it for other python versions either.