

# Bohr Sommerfeld quantization rule analysis using Matlab

Giacomo Boldrini 800692

## Abstract

Solving the Schrödinger equation analytically is not always possible therefore we need to find other methods in order to compute the eigenvalues of an Hamiltonian with a generic potential. Our job is to solve the eigenvalue equation for a potential  $|x|^\alpha$  with Matlab and to compare the results with the quasi-classical approximation of Bohr Sommerfeld.

## 1 Introduction

The WKB approximation (Wentzel-Kramers-Brillouin) is a method for finding approximate solutions to linear differential equations. In quantum mechanics it is typically used as a semi-classical approximation to the solutions of the Schrödinger's equation.

Since we do not care about the formal aspects of the theory we'll focus on the general results which lead us to our problem.

Let

$$\left( -\frac{\hbar}{2m} \nabla^2 + (U - E) \right) \psi = 0$$

be the time-independent Schrödinger equation (so  $\psi$  time independent) and  $\nabla^2 = \frac{\partial^2}{\partial x^2}$  the laplacian in one dimension.

Let's assume the solution to be

$$\psi = e^{\frac{i}{\hbar} \sigma} \quad \text{with} \quad \sigma = \sum_{i=0}^{\infty} \frac{\hbar}{i} \sigma_i = \sigma_0 + \frac{\hbar}{i} \sigma_1 + \dots \quad (*)$$

where  $\sigma$  is a function of the coordinates. Substituting  $(*)$  in the S. equation we get an equation for  $\sigma$ :

$\frac{(\sigma')^2}{2m} - i\hbar \frac{\sigma''}{2m} = E - U(x)$ . In first approximation we'll assume that the linear term in  $\hbar$  in  $(*)$  is negligible so that  $\sigma \simeq \sigma_0$ . We then have  $\frac{(\sigma_0')^2}{2m} = E - U(x)$  (1) which gives

$$\sigma_0 = \pm \int \sqrt{2m(E - U(x))} dx = \pm \int p(x) dx$$

where we used the classical expression for the momentum of the particle  $p = \sqrt{2m(E - U(x))}$ .

It turns out that the approximation (1) is valid only if  $|\frac{d}{dx}(\frac{\lambda}{2\pi})| \ll 1$  with the De-Broglie wavelength  $\lambda(x) = \frac{h}{p(x)}$ .

Doing manipulations the condition on the wavelength turns out to be equivalent to  $\frac{m\hbar|F|}{p^3} \ll 1$ . It is seen that the quasi-classical approximation becomes inapplicable if the momentum of the particle is too small. In particular, it is inapplicable near turning points i.e. near points where classically  $E = U(x)$  and  $p(x) = 0$ . At first order in  $\hbar$  we get  $\sigma_1 = \log(\frac{1}{\sqrt{p}})$ .

Therefore at 1st order we get the approximate solution as

$$\psi = \frac{C_1}{\sqrt{p}} \exp\left(\frac{i}{\hbar} \int p(x) dx\right) + \frac{C_2}{\sqrt{p}} \exp\left(-\frac{i}{\hbar} \int p(x) dx\right) \quad (2)$$

Let  $x=a$  be a turning point where  $E = U(a)$  and  $U(x) > E \quad \forall \quad x > a$ . For  $x > a$  the solution becomes

$$\psi = \frac{C}{2\sqrt{|p|}} \exp\left(-\frac{1}{\hbar} \int_a^x p(x) dx\right)$$

and for  $x < a$  we have (2). To determine the coefficients  $C_i$  we must follow the variation of the wave function around  $x=a$ . In doing so we would pass through a region where the quasi-classical approximation does not hold. This can be avoided by thinking  $\psi$  as a function of complex variable  $x$  that goes from positive to negative  $x - a$  along a path which is sufficiently far from  $x=a$ .

Going around  $a$  from right to left along a semicircle of large radius in the upper half-plane of complex variable  $x$  both  $U(x) - E$  and  $x - a$  acquire a  $\pi$  phase. Consequently the solution for the solution for  $x > a$  becomes the second term in (2) with a coefficient

$$C_2 = \frac{1}{2} C e^{-\frac{i\pi}{4}}$$

. Similarly on passing from right to left along a semicircle in the lower half-plane the function becomes the first term in (2) with coefficient

$$C_1 = \frac{1}{2} C e^{\frac{i\pi}{4}}$$

. Thus for  $x < a$  we get the solution as

$$\psi = \frac{C}{\sqrt{p}} \sin\left(\frac{1}{\hbar} \int_x^a p(x) dx + \frac{\pi}{4}\right) \quad (3)$$

In many potential such as  $|x|^\alpha$  we will have 2 turning points. Let's take the second turning point  $b$  such that  $b < a$  and  $U(x) > E$  for  $x < b$  and  $x > a$  and  $U(x) < E$  for  $b \leq x \leq a$  then

$$\psi = \frac{C'}{\sqrt{p}} \sin\left(\frac{1}{\hbar} \int_b^x p(x) dx + \frac{\pi}{4}\right) \quad (4)$$

Since (3) and (4) must be equal throughout the region, the sum of their phases must be an integer multiple of  $\pi$ :  $\frac{1}{\hbar} \int_b^a p(x) dx + \frac{\pi}{2} = (n+1)\pi$ . Rearranging and taking the integral over the whole period of classical motion we get

$$\oint p(x) dx = 2\hbar\pi(n + \frac{1}{2})$$

which is the Bohr-Sommerfeld quantization rule.

A more general quantization formula reads:  $\oint p(x) dx = 2\pi\hbar(n + \frac{1}{4} \sum_i \mu_i)$  where  $\mu_i$  is the Maslov correction, it's value is of the order of unity and depends on the boundary condition we impose on the wave function. The sum is over all the turning points.

## 2 Quantization for $U(x) = |x|^\alpha$

For the purposes of our problem we have to solve the quantization for the potential  $U(x) = |x|^\alpha$ . We have  $p = \sqrt{2m(E - |x|^\alpha)}$ . Suppose to fix the energy  $E$ . Since our potential is symmetric around  $x = 0$  we will have  $\forall \alpha$  two turning points  $x_2$  and  $x_1 = -x_2$  where  $E = U(x_1) = U(x_2)$ .

We have to compute

$$\int_{x_1}^{x_2} p(x) dx = \int_{x_1}^{x_2} \sqrt{2m(|x_2|^\alpha - |x|^\alpha)} dx = 2\sqrt{2m}(x_2)^{\frac{\alpha}{2}} \int_0^{x_2} \sqrt{1 - \left|\frac{x}{x_2}\right|^\alpha} dx$$

making the substitution  $\frac{x}{x_2} = t$  then  $dx = x_2 dt$  the integral becomes

$$2\sqrt{2m}(x_2)^{\frac{\alpha}{2}+1} \int_0^1 \sqrt{1 - |t|^\alpha} dt$$

we call the integral in  $t$   $C(\alpha) = \int_0^1 \sqrt{1 - |t|^\alpha} dt$  then we get  $2\sqrt{2m}(E)^{\frac{\alpha+2}{2\alpha}} C(\alpha) = \hbar\pi(n + \frac{1}{2})$  we finally get the Bohr-Sommerfeld quantization rule with  $\alpha$  free parameter:

$$E_n = \left[ \frac{\hbar\pi(n + \frac{1}{2})}{2\sqrt{2}C(\alpha)} \right]^{\frac{2\alpha}{2+\alpha}}$$

## 3 Numerical solutions with Matlab

The first thing we have to deal with using Matlab is that we are not in the continuous anymore. Our working space will be a grid of finite and equidistant points in one dimension, a vector. We take 1024 points and build the array `Ex` to have symmetry about  $x=0$ . We also define the potential on the elements of `Ex`:

```

N=1024;
Ex= -(N-1)/2:(N-1)/2;
Potential = @(x) abs(x).^a;
V = Potential(Ex);

```

We will then have to build the discrete Laplacian in order to compute the Schrödinger equation on the grid. We have various methods which differ in precision. The first one is the easiest. Let  $f$  be a function which takes values on the array. If  $a$  is the lattice spacing then  $f_j = f(ja)$  with  $j$  integer number and  $ja$   $j$ -th element of the array. We can take the definition of the derivative on the array with  $x = ja$  as  $\frac{df}{dx} = \frac{f(ja+a) - f(ja)}{a} = \frac{f_{j+1} - f_j}{a} = \nabla_+ f$  we can also define a negative shift  $\nabla_- f = \frac{f_j - f_{j-1}}{a}$ . With boundary conditions the derivative is an anti-hermitian operator but if we multiply the the complex  $i$  we get an hermitian operator  $\nabla_+^\dagger = -\nabla_-$ . We can introduce the finite difference laplacian as

$$-\nabla_+^\dagger \nabla_- = -\nabla^2 = \frac{2f_j - f_{j-1} - f_{j+1}}{a^2}$$

This way, the Laplacian of a function defined on the grid, only depends by the value of a point  $j$  and the two nearest points  $j-1$  and  $j+1$  therefore it is not the best approximation we can get for the operator. A better way to define the Laplacian on a discrete array is to work on the  $k$ -space which is connected to our  $x$ -space by the Fourier transform. It is a well-known theorem about Fourier transforms that

$$f'(x) = -ik\hat{f}(k)$$

where the l.h.s denotes the first derivative with respect to  $x$  and the r.h.s is the transform of  $f(x)$  in  $k$ -space multiplied by  $-ik$ .

The definition of Fourier transform is made on the continuous real axis (in one dimension) and it reads:

$$F\{f\}(k) = \hat{f}(k) = \frac{1}{\sqrt{2\pi}} \int_R f(x) e^{-ikx} dx$$

We can also define a discrete Fourier transform which takes values on an array for the functions. This is the result we will need. The discrete Fourier transform (DFT) transforms a sequence of  $N$  complex numbers  $\{x_n\}$  into another set of complex numbers  $\{X_k\}$  defined by:  $X_k = \sum_{n=0}^{N-1} x_n e^{\frac{-2\pi i}{N} kn}$ ;  $k = 0, 1, \dots, N-1$ . As we can see the value of the transform  $X_k$  depends on the value of all the components of the first array  $\{x_n\}$  so it's value is taken globally and not locally. Because of this reason, algorithms which can compute the Laplacian by DFT such as FFT are more precise. We will test two of them, the first is a redefinition of the finite difference Laplacian obtained with FFT.

The eigenvalue equation for the Laplacian in one dimension reads:  $\frac{\partial^2 u_k}{\partial x^2} = -k^2 u_k$  with periodical boundary conditions this is a simple differential equation with solutions plane waves  $u_k = \frac{1}{\sqrt{L}} e^{ikx}$   $k = \frac{2\pi}{L} n$   $n \in Z$ . Plane waves

are eigenfunctions of translations so we want to compute the finite difference Laplacian on these eigenfunctions. Let  $T$  be the translation operator on the grid and  $a$  the lattice spacing so  $\nabla^2 u_k = \frac{T+T^{-1}+2}{a^2} u_k = \frac{e^{ika}+e^{-ika}+2}{a^2} = \frac{2\cos(ka+2)}{a^2} = -\left(\frac{2}{a}\sin\left(\frac{ka}{2}\right)\right)^2$  so making the substitution  $-k \rightarrow -\tilde{k} = \frac{2}{a}\sin\left(\frac{ka}{2}\right)$  we should get the tri-diagonal Laplacian.

The second method just involves the  $k^2$  into the Fast Fourier transform and it's the most precise of the three.

We define  $k$

```
n = floor(N/2);
nn = floor((N-1)/2);
k = (2*pi/N)*(-n:nn)';
```

*floor* is a Matlab function which rounds the element of the array to the nearest integers. So  $n$  and  $nn$  are integers.

We then have to define the optimal lattice spacing  $dx$  in dependence of  $\alpha$ . Bigger  $\alpha$  requires a finer probing by our algorithm. Since  $T \sim a^{-2}$  and  $V \sim a^\alpha$  then  $\frac{T}{V} \sim a^{-2-\alpha}$  we want the lattice spacing to be proportional to  $\frac{1}{a}$  so we take:

$$dx = \left( \frac{\max(T)}{\max(V)} \right)^{\frac{1}{2+\alpha}} \sim \frac{1}{a}$$

We then redefine our array  $x$  and the potential  $V$

```
maxT=max(k.^2/2);
maxV=max(V);
dx = (maxT/maxV)^(1/(2+a));
x=Ex'*dx;
V = Potential(x);
```

We do the same thing using  $\tilde{k}$

```
maxT1 = max(4*sin(k/2).^2);
dx1 = (maxT1/maxV)^(1/(2+a));
x1 = Ex'*dx1;
V1 = Potential(x1);
```

We will introduce the dependencies of the  $k$  by  $dx$  while computing the FFT. Computing the kinetic energy matrix for  $\tilde{k}$ . We do not need to shift  $\tilde{k}$  while FFT because *sin* already has the required symmetry for the Fast Fourier but we will also need it later. The result of the FFT will be the  $-Laplacian$  matrix and we will divide it by  $2dx_1^2$  to obtain the kinetic energy matrix.

```
Z=fft(eye(N));
```

```

k1=fftshift(k);
T1=ifft(diag(4*(sin(k1/2)).^2)*Z);
T1 = real(T1+T1')/2;
T1(abs(T1)<1e-10)=0;
T1 = T1/dx1/dx1/2;

```

We do the same thing with  $k$  but we need to divide it by  $dx$  before computing the FFT. The result is directly the kinetic matrix:

```

k2=k1/dx;
T=ifft(diag(k2.^2/2)*Z);
T=real(T+T')/2;

```

The requirement  $T = \text{real}\left(\frac{T+T'}{2}\right)$  is the symmetrization condition. Since the potential will be a diagonal matrix we impose Hermiticity on  $T$  therefore the Hamiltonian will be an Hermitian matrix.

The finite difference Laplacian is a tri-diagonal matrix with the main diagonal of 2 (imposing periodic boundary conditions) and the upper and lower diagonals of -1. Requiring periodic boundary condition also means that  $L(1, N) = L(N, 1) = -1$ . We then divide by  $\frac{-1}{2}$  to obtain the kinetic energy.

```

eg = ones(N,1);
T2 = full(-0.5*spdiags([eg -2*eg eg],[-1 0 1],N,N) / dx1^2);
T2(1:10,1:10)

```

We define a function to obtain eigenvalues and eigenvectors called *diagonal*. We use the Matlab built-in function eig to diagonalize the Hamiltonian. The result of *diagonal* are the sorted eigenvectors and eigenfunctions.

```

function [u,e] = diagonal(T,V)
H=T+diag(V);
[u,e]=eig(H);
e=diag(e);
if ~issorted(e)
    [e,I]=sort(e);
    u(:,I);
end

```

We compute *diagonal* for our three different kinetic matrix and the same potential  $V$

```

[u,e]= diagonal(T,V); %k.^2/2
[u1,e1]= diagonal(T1,V1); %4sen(k/2).^2
[u3,e3]= diagonal(T2,V1); %Finite Difference

```

We compute the energy values with Bohr-Sommerfeld equation derived in the 2nd chapter:

```

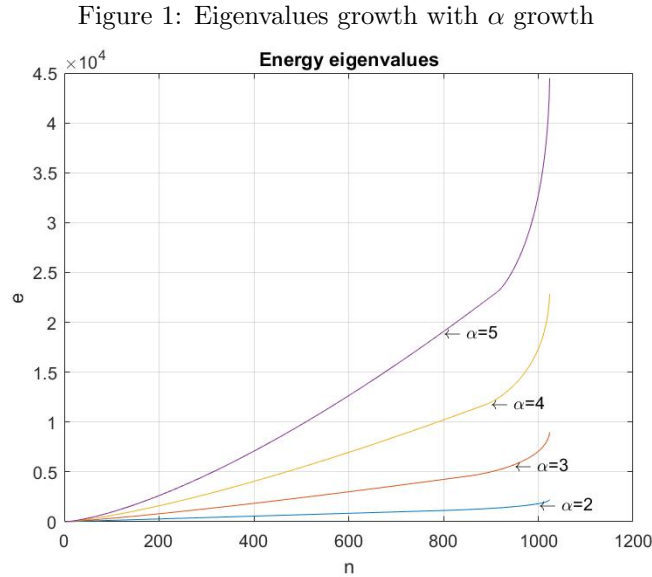
n=0:1:N-1;
En=((pi*(n+0.5))/(2*sqrt(2)*integral(a))).^(2*a/(a+2));

integral(a) is a function which computes the integral using symbolic expressions
 $\int_0^1 \sqrt{(1-|t|^\alpha)} dt$ 

function C = integral(a)
syms t;
y = sqrt(1-t.^a);
C = int(y,[0,1]);
end

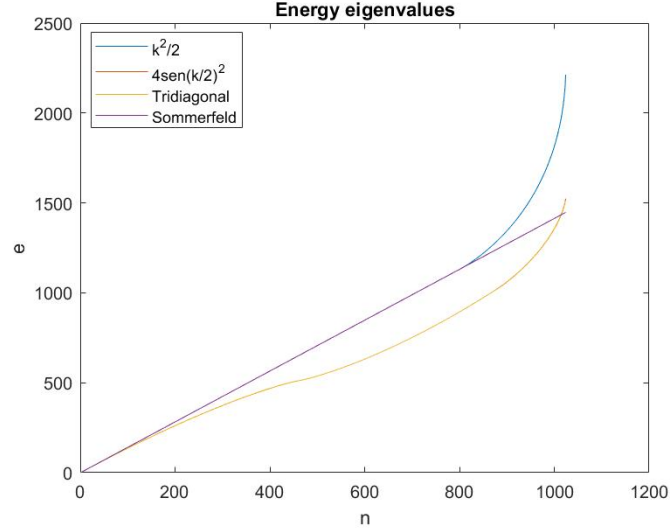
```

We plot the eigenvalues obtained from the diagonalization. We can first see the trend in the eigenvalues with the growth of  $\alpha$ :



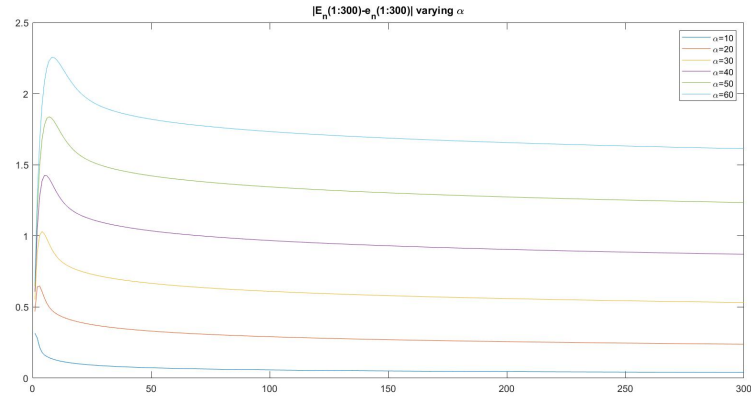
As we expect, eigenvalues with same  $n$  index are higher for higher  $\alpha$  since we have a bigger potential energy. Watching carefully we see something not expected: at a certain  $n^*$  (different for each  $\alpha$ ) the eigenvalues explode. We can see it better plotting the eigenvalues obtained from the diagonalizations with respect to the Bohr-Sommerfeld rule which we know it's true at least for  $\alpha = 2$  (harmonic oscillator):

Figure 2: Eigenvalues with FFT, finite difference and B-S  $\alpha = 2$



It's clear, from the plot, that the FFT with  $k$  gives the best approximated eigenvalues. We can test Bohr-Sommerfeld rule varying  $\alpha$  and plotting the absolute value of the difference between the eigenvalue from  $k$  FFT and B-S for the first 300 elements:

Figure 3: Difference in eigenvalues varying  $\alpha$

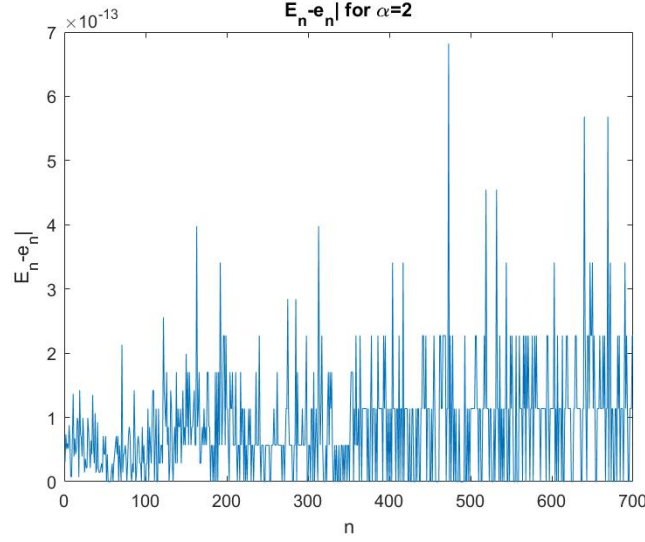


From this graph we can see that, at  $n$  fixed, the difference between Bohr-Sommerfeld and FFT eigenvalues is bigger with increasing  $\alpha$ . It's also evident that we have a similar trend, for the difference of the eigenvalues, for almost every  $\alpha$ .



Since we want to test the accuracy of Bohr-Sommerfeld it is worth asking if the relative high difference between the two eigenvalues is due to the FFT or to the quasi-classical approximation. In theory FFT gives the right approximated result for every  $\alpha$  since it is the diagonalization of the Hamiltonian (assuming that the approximations made by *eig* are negligible also for larger  $\alpha$ ). We can test this assumption on  $\alpha = 2$  where B-S is verified:

Figure 4: Difference in eigenvalues  $\alpha = 2$

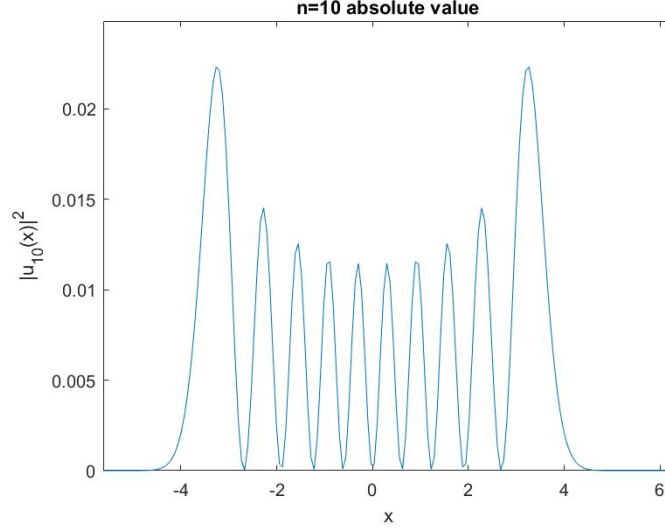


As expected even for large  $n$  the difference is negligible therefore FFT gives the right approximated eigenvalues for the potential (up to a certain quantum number where boundary problems arises as we will see). This can tell us that Bohr-Sommerfeld rule does not quite apply  $\forall \alpha$  but the error is also not that large.

As we said the eigenvalues computed with FFT with  $\tilde{k}$  are almost identical to the finite-difference ones. They are really good for small  $n$  but around  $n = 100$  the computation is wrong. The FFT computation with  $k$  instead is the best we can get and holds approximately up to  $n = 800$  with  $\alpha = 2$  then it diverges. This happens because of the not fine enough probing of our system. Since one of the outputs of our program gives the eigenfunctions we can plot the probabilities given as the square of the their absolute value. From the command line:

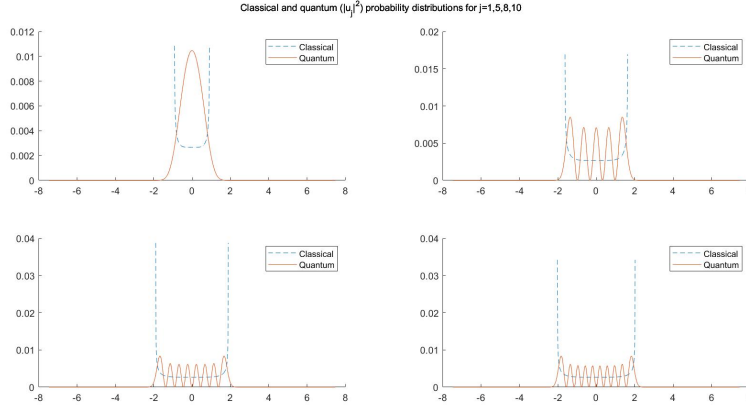
```
[u,~,~,x]=finale(2);
plot(x,(abs(u(:,10))).^2)
```

Figure 5: Probability of  $u_{10}$



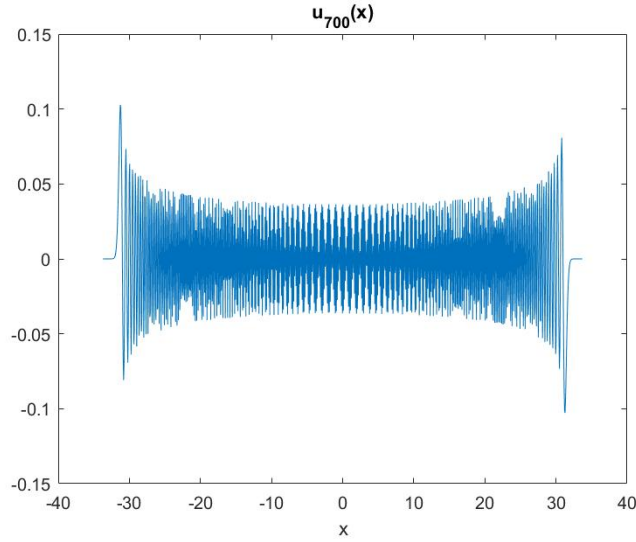
Classically we expect the probability to go like  $\frac{1}{p(x)}$  and we have integrable singularities at the turning points where  $p(x) = 0$ . Therefore we expect, for a particle, to spend more time on the turning points where its velocity is zero and the probability to find the particle in that region is higher. We can see the difference with quantum mechanics at the ground state (gaussian for an harmonic oscillator) where the highest probability is found in  $x = 0$  where the momentum is maximum. As  $n$  increase the maximum probability tends to be found at the turning points. At these points the eigenfunctions resent of the periodicity of the potential and interfere with themselves. For  $n \rightarrow \infty$  the zeroes of the functions are so near that we get a function that goes like the average of the spikes. The classical probability is defined as  $P_{cl} = \frac{dx}{v(x)T_{cl}}$  where  $T_{cl} = 2a\sqrt{\frac{m}{2E}}$  and is proportional to  $\frac{1}{p(x)}$

Figure 6: Classical and quantum probability distributions for  $u_j$   $j = 1, 5, 8, 10$   
 $\alpha = 5$



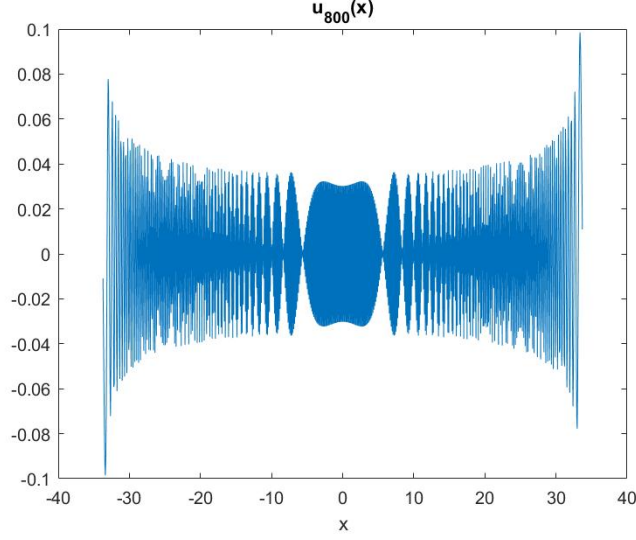
Furthermore for  $n \rightarrow \infty$  we expect the eigenfunctions to have  $n - 1$  zeros but our lattice is fixed so the zeroes become dense. For the computation to be correct we would like to have a smooth function over the lattice. If the step of the oscillations becomes comparable with the lattice spacing we are not probing fine enough the system and our computation of eigenfunctions and eigenvalues is wrong. That's the reason why FFT method does not work for large  $n$ . We can see this by plotting the eigenfunctions with  $n$  close to the point of divergence. As an example for  $\alpha = 2$ :

Figure 7:  $\alpha = 2$   $u_{700}$



We can see that the eigenfunction has many oscillations but still vanishes on boundaries as we expect. The difference between Bohr-Sommerfeld exact energy values and the computed one is  $En(700) - e(700) \sim -1.9650e - 13$ . Meanwhile, if we take  $n = 800$  we get:

Figure 8:  $\alpha = 2$   $u_{800}$

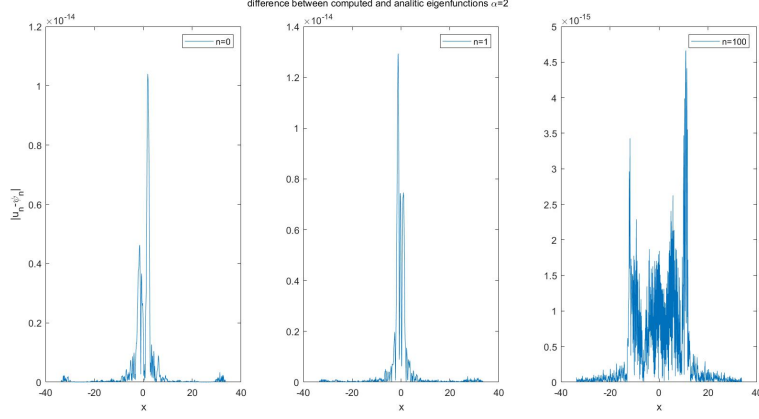


We see that the eigenfunction is not correct as it does not vanish on the boundaries in fact  $En(800) - e(800) \simeq 0.1681$  way larger than  $n = 700$ . Since for  $\alpha = 2$  the potential is an harmonic potential with  $\omega = \sqrt{2}$  we can plot the difference between the analytic solutions and the numerical solution to test the precision of our algorithm. The exact solutions of the harmonic oscillator are:

$$\psi_n(x) = \frac{1}{\sqrt{2^n n!}} \left( \frac{m\omega}{\pi \hbar} \right)^{\frac{1}{4}} e^{-\frac{m\omega x^2}{2\hbar}} H_n\left(\sqrt{\frac{m\omega}{\hbar}} x\right) \quad n = 0, 1, 2, \dots$$

where  $H_n$  are the Hermite polynomials. The function in Matlab `hermiteH(n, x)` represents the  $n$ th-degree Hermite polynomial at the point  $x$ . The result is:

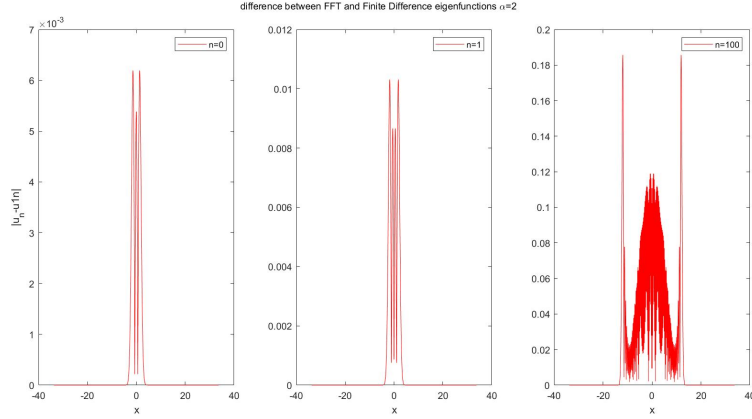
Figure 9: difference in numerical and analytic solutions  $|u_n(x) - \psi_n(x)|$



We can see that the scale of the difference is negligible. The fact that for  $n = 100$  we have a smaller difference does not imply that for larger  $n$  the algorithm is more precise, it just tells us that up to  $n \simeq 800$  the difference is negligible and we'll have larger fluctuations across the whole lattice. We also have a small fluctuation around the turning points.

If we plot the difference between the FFT and the 'finite difference' eigenfunctions we can see that, as the respective eigenvalues tends to diverge, we get larger differences in the eigenfunctions:

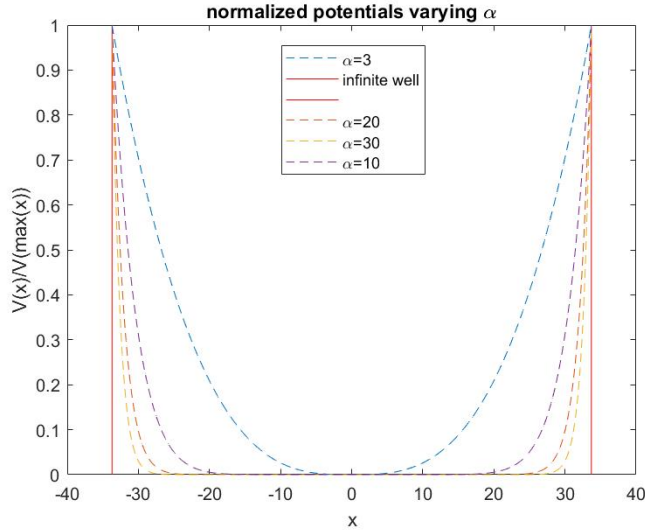
Figure 10: difference in numerical solutions  $|u_n(x) - u1_n(x)|$



## 4 Infinite potential well

We have problems when  $\alpha$  is a number higher than 100 (we can actually compute up to  $\alpha = 113$ ). Our array  $x$  takes values in the range  $[-511.5; 511.5]$  if we want to compute the potential for  $\alpha = 113$  we would get  $(1.26)10^{306}$  which is really close to the largest positive floating-point number in Matlab *realmax*  $\simeq (1.79)^{308}$ . For  $\alpha = 114$  we would get larger numbers than *realmax* and Matlab gives error message. If  $\alpha$  is large we expect to have an infinite potential well where the potential is approximately infinite at the boundaries and zero inside our region of interest. We can see that this approximation holds if we plot the graphs of the potentials with different values of  $\alpha$  divided by their largest value on our grid (since the potential is symmetric this will be one of the ends of our array  $x$ ) to compare the values of the potential at the boundaries and on the inside, their ratio:

Figure 11: Ratio  $\frac{V(x)}{V(\max(x))}$ , comparing boundaries and inner values



In fact we can also see this mathematically:  $\lim_{\alpha \rightarrow \infty} |x^\alpha| = \begin{cases} 0 & \text{if } x \in (-1, 1) \\ \infty & \text{elsewhere} \end{cases}$

We expect, for large  $\alpha$ , our eigenvalues to be the ones of a potential well defined in  $x \in (-1, 1)$ .

At the beginning of our program we have a condition on  $\alpha$ .

```
if a > 100
    disp('a is too large for Matlab to compute. We approximate to potential well')
    finale1;
    return
```

end

This condition calls another program, *finale1*, if the input is bigger than 100 and terminates the execution of the current program.

As the case with  $\alpha = 2$ , in the infinite potential well the quasi-classical approximation of Bohr-Sommerfeld gives the exact result.

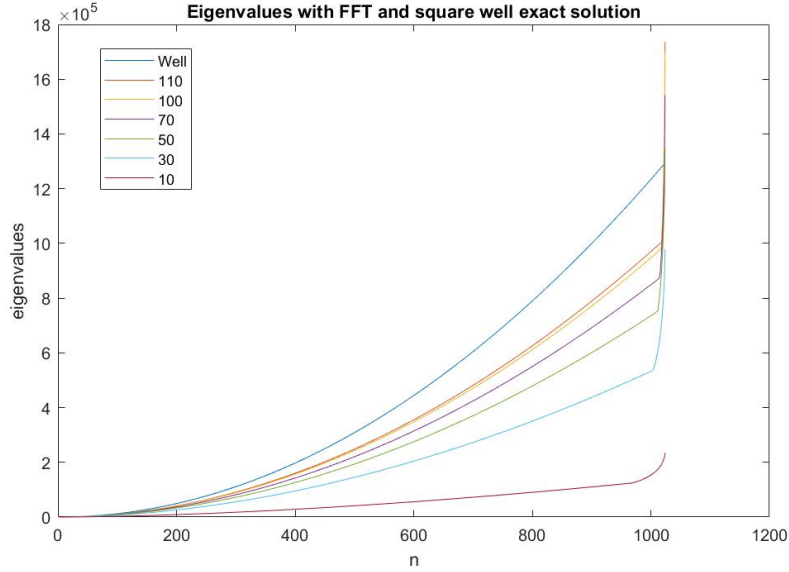
The turning points now sit at the edges of our lattice interval of length  $L = Ndx$ .  $U(x) = 0 \quad \forall x$  in the the lattice and  $U(x) = \infty$  outside. Therefore  $\oint p(x)dx = nh \quad n = 1, 2, \dots$  note that we do not have the additional  $\frac{1}{2}$  to the quantum number  $n$  because at the edge of the well  $E$  should be infinite which is not physically possible.  $2 \int_0^L p dx = 2Lp$  since  $p = \sqrt{2mE}$  does not have  $x$  dependence. Solving for  $E$  we get:

$$E_n = \frac{n^2 \pi^2 \hbar^2}{2mL^2} \quad n = 1, 2, \dots$$

Which are the exact solutions of the infinite potential well.

If we have  $x \in (-1, 1)$  then  $L = 2$ . We can see the trend of the eigenvalues of  $|x^\alpha|$  with growing in  $\alpha$ :

Figure 12: Trending toward the infinite potential well as  $\alpha$  increases



As expected we have a trend toward the eigenvalues of the infinite well. We still can't compute eigenvalues for  $\alpha > 113$  but we assume that the system for such values is just the infinite potential well.

Since the potential is infinite outside our region we require Dirichlet boundary

conditions at the edge of our lattice. This gives a discretization on the wavenumber  $k$ . The Schrödinger equation is the same as the Laplacian eigenvalue equation  $\frac{\partial^2 u_k}{\partial x^2} = -k^2 u_k$  but with the requirement that  $u_k(-\frac{L}{2}) = u_k(\frac{L}{2}) = 0$ . The general solution reads as  $u_k(x) = A \sin(kx) + B \cos(kx)$  and with DBC we get two class of functions eigenstates of parity:  $u_k^{odd}(x) = \sqrt{\frac{2}{L}} \sin(k_n x)$  for  $n = 2, 4, \dots$  and  $u_k^{even}(x) = \sqrt{\frac{2}{L}} \cos(k_n x)$  for  $n = 1, 3, \dots$  given the quantized wavenumber  $k_n = \frac{\pi}{L} n$   $n = 1, 2, 3, \dots$ .

In *finale1* we do exactly the same procedure of *finale*, but this time we define the lattice as a box  $(-1,1)$  with  $N = 1024$  points spaced by  $dx = \frac{2L}{N}$  where  $L = 1$  then our new  $k$ :

```
L=1;
N=1024;
dx = 2*L/N;
x = -L*(1-1/N) : dx : L*(1-1/N);
k = (pi/dx/N)*(1:N)';
```

Since we do not have periodic boundary conditions but Dirichlet boundary conditions we'll use *sinft*, a sine Fourier transform (DST), to compute the kinetic matrix. The difference between DFT (Discrete fourier transform) and DST is that the second only uses sines functions while the first also uses cosines.

```
T = sinft(diag(k.^2/2)*sinft(eye(N)));
T = real(T+T')/2;
```

Since  $U = 0$  we obtain eigenvalues and eigenfunctions of  $T$ :

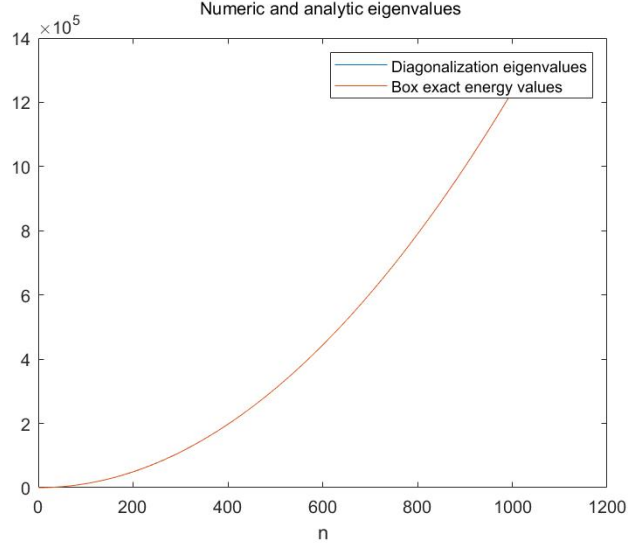
```
[u,e]=eig(T);
e=diag(e);
if ~issorted(e)
    [e,I]=sort(e);
    u(:,I);
end
```

We compare the results of the diagonalization with the exact eigenvalues:

```
En = ((1:N).^2)*pi^2/(2*(dx*N)^2);
```

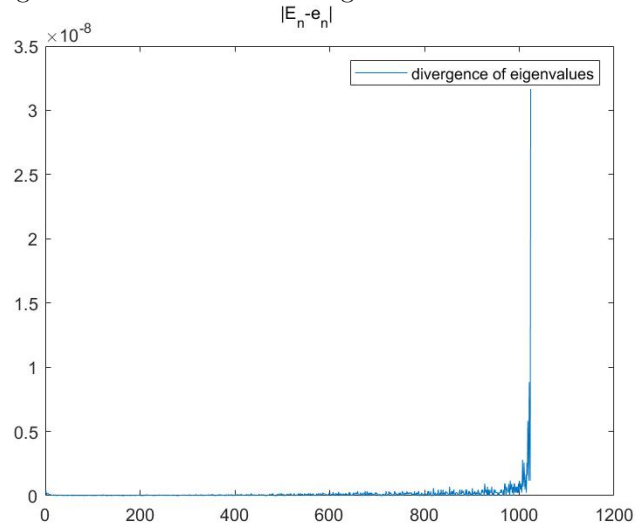


Figure 13: Comparing numerical and analytic eigenvalues



The two plots are indistinguishable therefore we can plot the difference of the two results:

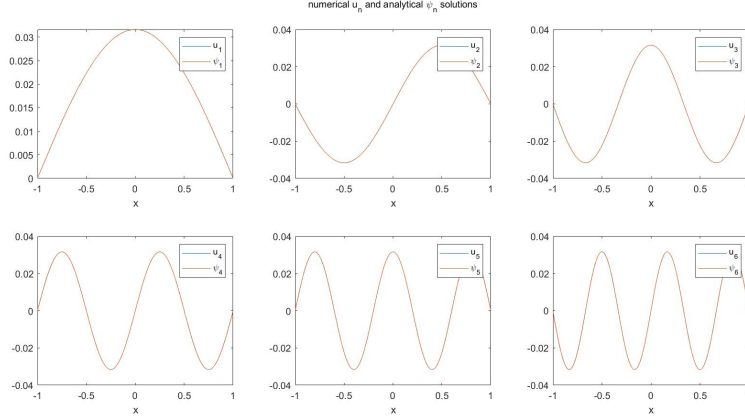
Figure 14: Difference of the eigenvalues as function of  $n$



As expected the difference is negligible since Bohr-Sommerfeld actually gives the exact energy values for the infinite potential well and the DST method gives a precise approximation.

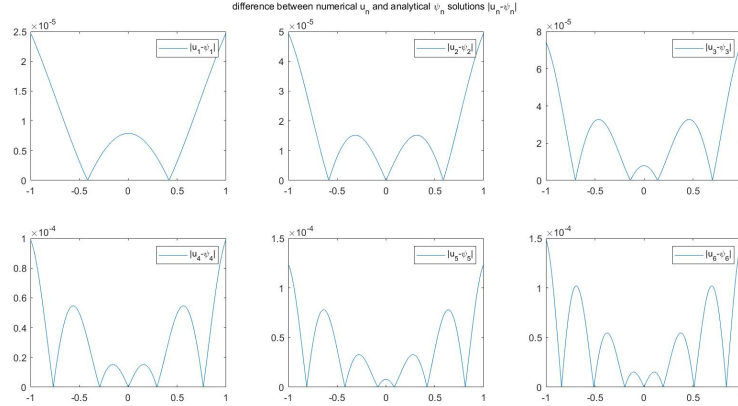
As in *finale* we plot analytical and computed eigenfunctions since we know the exact solutions. We then plot the difference to estimate the error of our computation:

Figure 15: numerical  $u_n$  and analytical  $\psi_n$  solutions



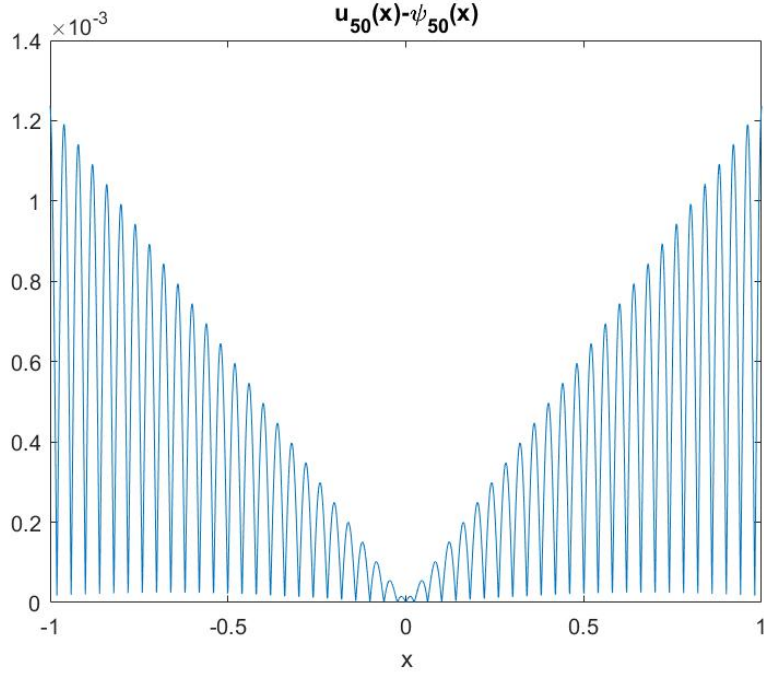
The two functions are always indistinguishable when plotted. The difference is of the order of  $10^{-5}$  for the first quantum numbers and has it's higher value in correspondence of the turning points:

Figure 16: Difference between numerical and analytic eigenfunctions



We can see that the order of magnitude of the difference tends to increase as  $n$  increase. As an example we plot  $|u_{50} - \psi_{50}|$ :

Figure 17: Difference for large  $n$



The fact that the difference between the computed and the analytic functions is large for bigger  $n$  is due to the fact that we have a fixed lattice spacing. As soon as the function is not smooth on the lattice spacing (as  $n \rightarrow \infty$  we have more zeros therefore more oscillations of the eigenfunction) our computations will give an erroneous result.

## 5 Conclusions

Even if the Bohr-Sommerfeld quantization rule was, in principle, an ad hoc. formula to get the exact energy spectrum of the hydrogen atom it still gives good approximated results for potentials of the type  $|x|^\alpha$ . The fact that for  $\alpha = 2$  (harmonic oscillator) it gives the exact energy eigenvalues is just a coincidence and the rule does not apply to all potentials. A well known example where one of these deficiencies appear is the radial Schrödinger equation with a Coulomb singularity in the potential at the origin and with  $l \neq 0$ . In this case, the WKB half-integer quantization condition does not, in general, yield the correct energy levels.

Anyway the WKB approximation satisfy the Bohr correspondence principle so for  $n \rightarrow \infty$  we should get the classical values of the energy (we can't compute them due to FFT inaccuracy as for  $n \rightarrow \infty$  we get oscillations steps comparable with the lattice spacing but in theory Bohr-Sommerfeld and FFT eigenvalues

should coincide). We can see this in two ways: first of all the probability expressed as the square of the absolute value of the wavefunction computed with FFT methods tends  $\forall \alpha$  as  $n \rightarrow \infty$  to align with the classical probability. Furthermore, for  $\alpha \neq 2$  we see that the trend in the difference of the computed and quantized eigenvalues is the same, has a maximum for small  $n$  and tends to decrease as  $n$  increases and should give the exact classical result for large  $n$  (Bohr correspondence principle).