Solving the Time Independent Schrödinger Equation on a Lattice

Max Heimann 580560, Lyding Brumm 604522, Burak Varol 623941

November 2023

Contents

1 Introduction

Solving the time independent Schrödinger equation (TISE) is only possible in an analytical fashion for a subclass of trivial problems. This explains the outstanding importance of analytical solvable potentials like e.g. the quantum harmonic oscillator, but wherever there is no suitable model potential a solution can be still obtained by numerical calculations. We present an algorithm to obtain the ground-state and its energy by approximating the continuous case by a discrete lattice. We prove some useful properties of the discretised Hamiltonian and the use algorithm to solve the 'Mexican hat' potential:

$$V(x) = \frac{m\omega^2}{8r^2}(x^2 - r^2)^2$$

which will be completely described in terms of natural units. The code consists of following modules:

Module Name	Summary
grid.py	Implements grid related methods
hamilton.py	Implements the hamiltonian
eigen.py	Implements the power-methode and conjugate-gradient algorithm
observables.py	Implements a set of operators
main.py	The main code and cli-based user interface
qho.py	A toy example to test nondimensionalisation
1d_calcs_and_vis.py	Visualise 1d potential and wave-function
2d_calcs_and_vis.py	Visualise 2d potential and wave-function

Table 1: Summary of python modules

2 Modules

2.1 Grid Module

This module creates:

- The grid that we want to work with. In this code every dimension is discretised in $2N+1$ points, where the points span from $-N, .., 0, ...N$, with total D dimensions. To create the grid we use the numpy function "np.meshgrid" and then stack the coordinates together with "np.stack" to create a human readable grid.
- The grid which contains the distance squared values for every value. This part is the main part that we need to calculate the potential.

2.2 Hamilton Module

This Module contains:

- The potential energy operator. The function calls "calculate grid distance square" function from grid module and applies the formula $V = \frac{\mu}{8}(\epsilon^2|n|^2 - 1)^2$. Result is an np.ndarray.
- The kinetic energy operator. The function applies the formula $-\frac{1}{2\mu\epsilon^2}\Sigma_k(\psi(n + e_k) + \psi(n - e_k) - 2\psi(n))$. Here we use the function "np.roll" which considers the values that are translated in the specified direction when called. This imposes periodic boundary condition.
- The create _ hamiltonian function, which returns the Hamiltonian. We created this so that we can create the Hamiltonian operator and save it to a variable for further use.

2.3 Eigenvectors/values Module

This Module consists of:

- "get highest eigenvalue" and "apply inverted" functions from warm-up exercises, details can be found in the related project. We slightly modified them by creating "gen invert operator", where now the parameters "tolerance" and "res recal iterations" are directly passed to apply inverted; and "get smallest eigenvalue" function, where the smallest eigenvalue are calculated by calculating the highest eigenvalue of operator^{-1}.

- Three test functions "test get highest eigenvalue", "test get smallest eigenvalue" and "test apply inverted." The first two creates random matrices with positive determinants and calculates the highest/lowest eigenvalue with corresponding eigenvector. If the calculated values are true, then $\langle v^T A v \rangle$ is the eigenvalue and $1 - \frac{\langle v^T A v \rangle}{\lambda}$ is the error, where λ is the true eigenvalue. Actually, this just answers the question if the calculated values are eigenvalues or not, but due to the nature of the algorithm it must be the highest/lowest eigenvalue. The last test compares the result with the multiplication of the proper inverse matrix multiplied to a test vector.

2.4 Observables

This Module consists of the functions:

- "mu O", calculating the expectation value of an observable O
- "var O", calculating the variance of an observable O
- "stat psi" where it calculates the upper two at the same time. If individual values are needed then it is more efficient to call one of the first functions. If both values are needed , the third function would be faster.
- "apply x 1d" Applies the location operator x along an axis to psi.
- "sphere r prob" finds the probability to find a the particle within a radius of r for a given wave-function.
- " single slice " which slices an array at a specific axis. (not used, but kept for convenience)
- "mult along axis" which multiplies a 1d array (B) along an axis of an Nd-array(A). This is helpful for the location operator

Tests for Gaussian

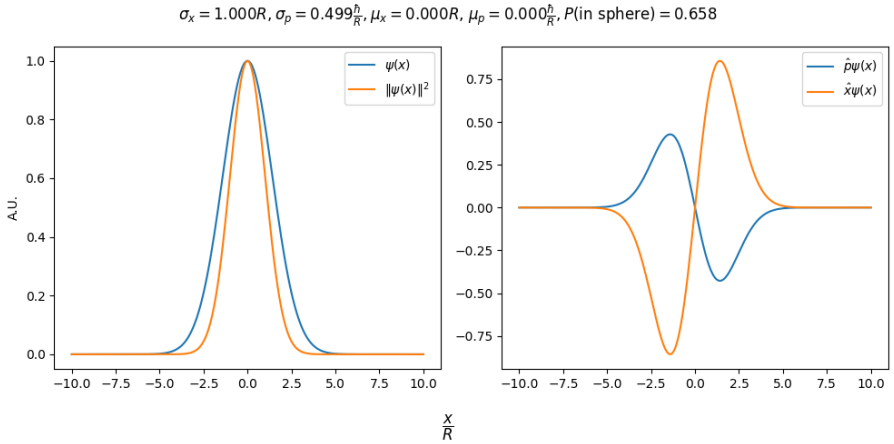

Figure 1: Observables of a Gaussian test function

To run a quick test on the observables module we run every function on a wave-function with a Gaussian PDF which has $\sigma = R$ in 1D. This way we can do a easy sanity check and see if everything works as expected, i.e. the expectation values for $\hat{x}$ and $\hat{p}$ are zero as well as $\sigma_x = R$ and $\sigma_p = \frac{\hbar}{R}$. Additionally, the probability of the particle being inside the sphere of radius R (the interval $[-R, R]$) is about 65.8%. Everything works as expected.

Tests of the modules

Each module comes with its own test code, which is run if the module invoked by 'python $module_name'

Listing 1: Output of hamilton.py

```
Error in linearity:  2.233716114265519e-21
Smallest <phi|H|phi>:  0.14346846666556545
Error in hermiticity:  4.656612873077393e-10
Error in planar wave eigenvectorness:  (3.477988774048796e-17+0j
```

The error of linearity is in the order of the double precision floating point number.
The smallest energy expectation value of arbitrary wave functions has to be larger than zero.
The error in hermiticity is orders of magnitudes worse than the error in linearity, which is due to discontinuities of the random test vectors we used. This way the floating point uncertainties of subtracting large and small numbers add up and explain the increased error. We also used continuous test vectors and indeed got smaller errors ($\approx 10^{-14}$).
The error of planar wave 'eigenvectorness' is again orders of magnitudes higher than the precision of double floating point numbers, but the increase is explained by the approximation of finite differences.

Listing 2: Output of eigen.py

```
test get highes eigenvalue:  4.394262784136936e-13
test apply_inverted:  [[2.54412328e-07-6.10760215e-07j]]
test get lowest eigenvalue:  1.6936374258591513e-07
```

Here the tests where run with a tolerance of 10^{-6}, all errors are smaller than the specified tolerance.

2.5 Usage of the main code

The module "main.py" implements a cli-based user-interface for the our TISE-solver. All following simulation parameters have to be specified in an input file:

Parameter Name	Summary
eps	First simulation parameter
mu	Second simulation parameter
N	The number of discretisation points
D	The dimension of the grid
tolerance_cg	The acceptable error for conjugate gradient
tolerance_pm	The acceptable error for power-method
res_recal	How frequently should the residuum recalculated precisely
groundstate_output_file	Where to save the wave-functions

Table 2: Description of input parameters

The program will calculate the ground-state energy, the expectation value of the location, expectation value of the momentum, their standard deviations and write the back to the input file. The wave-function will be stored in a numpy binary format, so it can be processed easily in the future. Multiple input files can be specified and will be processed sequentially, implementing multiprocessing support should be fairly easy and might be added in the future.

3 System of Natural Units

Over the centuries of physical research different branches of physics prefer different systems of units. This is known to be a source of errors. Even if there are modern attempts to incorporate unit systems to programming languages it is still considered to be best practice to describe each physical system in its own units. Is the system described by a Hamiltonian i.e. a kinetic energy and a potential energy term the potential dictates a system of units. For the 'Mexican hat' potential the parameters are:

$$\mu = \frac{m\omega r^2}{\hbar}$$
$$\epsilon = \frac{a}{r}$$

Where a is the lattice constant.

4 The discretised Hamiltonian and its properties

In continuous quantum mechanics hermiticity of all operators corresponding to observables is an axiom. This is therefore true for the Hamiltonian of a system. After discretisation hermiticity and other properties of the Hamiltonian operator has to be proven. The scalar product on a finite dimensional Hilbert space is defined as:

$$(\psi, \phi) = \sum_n \psi(n)^* \phi(n)$$

The Hamiltonian is defined as:

$$H_a\psi(n) = -\frac{1}{2\mu\epsilon^2}\sum_k [\psi(n+e_k) + \psi(n-e_k) - 2\psi(n)] + \frac{\mu}{8}(\epsilon^2 n^2 - 1)^2\psi(n)$$

and therefore can be split up into a kinetic and potential term. This allows use to prove linearity and hermiticity of both terms separately which makes the proves shorter. We will now prove these properties.

Theorem 4.1. *Let V be the discretized unitless (real) potential operator. Then $V(\alpha\psi + \beta\phi) = \alpha V\psi + \beta V\phi$, where ψ and ϕ are valid wave-functions and $\alpha, \beta \in \mathbb{C}$, i.e. V is linear.*

Proof.

$$V(\alpha\psi + \beta\phi) = \sum_n V(n)(\alpha\psi(n) + \beta\phi(n)) =$$
$$\sum_n V(n)\alpha\psi(n) + V(n)\beta\phi(n) = \sum_n \alpha V(n)\psi(n) + \sum_n \beta V(n)\phi(n) =$$
$$\alpha\sum_n V(n)\psi(n) + \beta\sum_n V(n)\phi(n) = \alpha V\psi + \beta V\phi$$

where $V(n) = \frac{\mu}{8}(\epsilon^2 n^2 - 1)^2$ □

Theorem 4.2. *Let T be the discretized unitless kinetic energy operator. Then $T(\alpha\psi + \beta\phi) = \alpha T\psi + \beta T\phi$, where ψ and ϕ are valid wave-functions and $\alpha, \beta \in \mathbb{C}$, i.e. T is linear.*

Proof.

$$T(\alpha\psi + \beta\phi) =$$
$$\frac{1}{2\mu\epsilon^2}(\sum_n 2(\alpha\psi + \beta\phi)(n) - \sum_k (\alpha\psi + \beta\phi)(n+e_k) - \sum_k (\alpha\psi + \beta\phi)(n-e_k)) =$$
$$\frac{1}{2\mu\epsilon^2}\sum_n 2\alpha\psi(n) + 2\beta\phi(n) - \sum_k \alpha\psi(n+e_k) + \beta\phi(n+e_k) - \sum_k \alpha\psi(n-e_k) + \beta\phi(n-e_k) =$$
$$\alpha\left(\frac{1}{2\mu\epsilon^2}\sum_n 2\psi(n) - \sum_k \psi(n+e_k) - \sum_k \psi(n-e_k)\right) + \beta\left(\frac{1}{2\mu\epsilon^2}\sum_n 2\phi(n) - \sum_k \phi(n+e_k) - \sum_k \phi(n-e_k)\right)$$
$$= \alpha T\psi + \beta T\phi$$

□

Corollary 4.2.1. *Since V and T are linear their sum also is, i.e. H is linear.*

Theorem 4.3. *Let ϕ, ψ be wave-functions and V the real potential operator, then $(\psi, V\phi) = (V\psi, \phi)$, i.e. V is Hermitian.*

Proof.

$$(\psi, V\phi) = \sum_n \psi(n)^* V(n)\phi(n) = \sum_n \psi(n)^* V(n)^* \phi(n) = \sum_n V(n)^* \psi(n)^* \phi(n) = (V\psi, \phi)$$

□

Theorem 4.4. *The kinetic energy operator is hermitian.*

Proof. Note that $\delta_{i,j+k} = \delta_{j,i-k}$ We use the orthonormal position basis $\mathcal{B}$. We can write:

$$T_{nm} = e_n^\dagger T e_m = \frac{1}{2\mu\epsilon^2}\sum_k (2\delta_{n,m} - \delta_{n,m-e_k} - \delta_{n,m+e_k})$$
$$= \frac{1}{2\mu\epsilon^2}\sum_k (2\delta_{m,n} - \delta_{m,n+e_k} - \delta_{m,n-e_k}) = e_m^\dagger T e_n = T_{mn}$$

So the kinetic operator is symmetric. As it is purely real as well, the statement is shown. □

Corollary 4.4.1. *Since V and T are hermitian their sum also is, i.e. H is hermitian.*

Theorem 4.5. *V is semipositive definite, i.e. $(\psi, V\psi) \geq 0 \ \forall\psi$*

Proof.

$$(\psi, V\psi) = \sum_n \psi(n)^* V(n)\psi(n) = \sum_n V(n)\psi(n)^*\psi(n) \geq 0$$

Since $V(n) \geq 0 \ \forall n$. □

Note: $V(n) = 0$ is only true if $n^2 = \frac{1}{\epsilon^2}$. From that follows:

$$\{\psi \,|V\psi = 0\} = \{\psi = \sum_{n^2=1/\epsilon^2} \alpha_n e_n \,|\, \alpha_n \in \mathbb{C} \,\forall n\}$$

For the next proof we will need a well known fact from linear algebra: all diagonally dominant matrices are semipositive, i.e. if $(A)_{ii} \geq \sum_{j\neq i} |(A)_{ij}|$ then A is semipositive. This may be looked up in any decent linear algebra textbook.

Theorem 4.6. *T is semipositive, i.e. $(\psi, T\psi) \geq 0 \ \forall\psi$*

Proof. We will again use the position basis $\mathcal{B}$ to represent T as a matrix.

$$T_{nn} = \sum_k 2 = 2D, \qquad \sum_{n\neq m} |T_{nm}| = \sum_{m\neq n}\sum_k (-2\delta_{m,n} + \delta_{m,n+e_k} + \delta_{m,n-e_k})$$

$$\sum_{m\neq n}\sum_k (-2\delta_{m,n} + \delta_{m,n+e_k} + \delta_{m,n-e_k}) = \sum_k \sum_{m\neq n} (-2\delta_{m,n} + \delta_{m,n+e_k} + \delta_{m,n-e_k}) =$$

$$\sum_k 2 = T_{nn}$$

□

We now only need one more ingredient to show that H is semipositive.

Lemma 4.7. *The sum of two semipositive linear operators A and B is also semipositive.*

Proof.

$$(x, (A+B)x) = (x, Ax + Bx) =$$
$$(x, Ax) + (x, Bx) \geq 0$$

□

Corollary 4.7.1. *Since T and V are both semipositive H also is semipositive.*

The eigenfunctions of the continuous kinetic energy operator are plane waves. Restricting the overall space to a box with periodic boundary conditions imposes a restriction to the k-vector of the planar wave functions as known from e.g. Solid State Physics. The same is true for our discretisation:

Theorem 4.8. *The discretised plane wave-functions $\phi_k(n) = \exp\left(\frac{2\pi i n k}{N}\right)$ are eigenfunctions of the kinetic energy operator. Where $k \in \mathbb{Z}^D$.*

Proof. First one notices that the (discretised) plane wave-functions are product functions of 1d plane wave-functions.

$$\phi_k(n) = \prod_{l=1}^{D} \exp\left(\frac{2\pi i n_l k_l}{N}\right)$$
$$= \exp\left(\frac{2\pi i n_l k_l}{N}\right)\xi_l$$

where ξ_l is the product over all $1d$ plane wave functions with indices not equal to l. Obviously $\left[T_l^{1D}, \xi_l\right] = 0$, where T_l^{1D} denotes the 1d kinetic energy operator in direction of l. With this one can write:

$$\begin{aligned} T\phi_k(n) &= -\frac{1}{2\mu\epsilon^2}\sum_l \left(2\exp(2\pi i n_l k_l/N) - \exp(2\pi i(n_l+e_l)k_l/N) - \exp(2\pi i(n_l-e_l)k_l/N)\right)\xi_l \\ &= \frac{1}{2\mu\epsilon^2}\sum_l \left(\exp(2\pi i n_l k_l/N)\exp(-2\pi i k_l/N) + \exp(2\pi i n_l k_l/N)\exp(+2\pi i k_l/N) - 2\exp(2\pi i n_l k_l/N)\right)\xi_l \\ &= \frac{1}{2\mu\epsilon^2}\sum_l \left(\exp(-2\pi i k_l/N) + \exp(+2\pi i k_l/N) - 2\right)\exp(2\pi i n_l k_l/N)\xi_l \\ &= \frac{1}{\mu\epsilon^2}\left(\sum_l 1 - \cos(2\pi k_l/N)\right)\phi_k(n) = \frac{2}{\mu\epsilon^2}\left(\sum_l \sin^2\left(\frac{\pi e_l k_l}{N}\right)\right) \end{aligned}$$

□

Corollary 4.8.1. *ϕ_0 has an associated eigenvalue of 0. Since there can only be N^D independent eigenvectors, it is also the only eigenvector with this eigenvalue, as the others N^D-1 vectors are the rest of the ϕ_k corresponding to the inverse lattice points which are not 0 and their eigenvalues are not 0.*

We can now go even further and show that H is in fact strictly positive. For this we will show two more things:

1. If A is semipositive and hermitian, then the all x that fulfil $(x, Ax) = 0$ are within the span of the eigenvectors with eigenvalue 0.
2. The sum if two semipositive linear operators A and B is strictly positive if their respective sets of vectors that fulfil $(x, Ax) = 0$ and $(x, Bx) = 0$ are mutually exclusive

We will introduce some notation to make our lives easier.

$$\{x\,|(x, Ax) = 0\} = \kappa_A$$

Theorem 4.9. *Let A be semipositive and hermitian. Then* $\mathrm{Span}(\{x\,|Ax = 0\}) = \kappa_A$

Proof. Obviously, if A is semipositive all, its eigenvalues are bigger than or equal to 0. So if x_i is an eigenvector with eigenvalue λ_i

$$(x_i, Ax_i) = (x_i, \lambda_i x_i) = \lambda_i \|x_i\|^2 \geq 0 \Rightarrow \lambda_i \geq 0$$

Now since A is hermitian, all its eigenvectors are orthogonal and can be normalized. Since x_i is a complete orthonormal basis we can represent every vector as a linear combination of these eigenvectors. Now let us look at a vector $y = \sum_i \alpha_i x_i \in \kappa_A$

$$0 = (y, Ay) = (\sum_i \alpha_i x_i, A\sum_j \alpha_j x_j) =$$

$$\sum_{i,j} \alpha_i^* \alpha_j (x_i, \lambda_j x_j) = \sum_{i,j} \lambda_j \alpha_i^* \alpha_j \delta_{ij} = \sum_i \lambda_i |\alpha|^2$$

Now this sum can only become zero if every summand is zero. This means for $\lambda_i = 0$, α_i can really be any number. For $\lambda_i \neq 0$, α_i must be 0 for y to be in κ_A. □

Theorem 4.10. *Let A and B be semipositive. Then $\kappa_{A+B} = \kappa_A \cap \kappa_B$.*

Proof. Since A and B is semipositive, $A + B$ also is. Now let $x \in \kappa_{A+B}$

$$\begin{aligned} &0 = (x, (A+B)x) = (x, Ax) + (x, Bx) \\ \iff\ &0 \geq -(x, Bx) = (x, Ax) \geq 0 \end{aligned}$$

Now the above equation can only be true if $x \in \kappa_A \cap \kappa_B$. The inverse is trivial:

$$0 = (x, Ax) + (x, Bx) = (x, (A+B)x)$$

□

Corollary 4.10.1. *$A + B$ is strictly positive if $\kappa_A \cap \kappa_B = \{\}$*

We have already shown that T and V are hermitian, meaning κ_V and κ_T are the spans of the respective eigenvectors with eigenvalue 0. Now the last thing left to prove, is that ϕ_0 (the only vector in κ_T) is not in κ_V. Therefore let us choose some vector from the positional basis e_m which is not in κ_V, so any point for which $m^2 \neq 0$. Now obviously

$$(e_m, x) = 0 \quad \forall x \in \kappa_V$$

must be fulfilled. So does this hold true also for ϕ_0?

$$(e_m, \phi_{k=0}) = \sum_n (e_m, \mathrm{e}^{0*2\pi i/N} e_n) = 1 \neq 0$$

No! This means that ϕ_0, the only vector in κ_T is not in κ_V. By our last corollary that implies H is positive. Note that this actually implies that all discrete Hamiltonians are positive, as long as V is semipositive and not zero somewhere.

5 Results

5.1 Script values

We ran the program for 1D and 2D for the given parameters in the script. The results are visualized in Figures 2 to 4. The observalbles for the 1D case are

$$\mu_{\vec{x}} = -0.0780r \qquad \mu_{\vec{p}} = 0.0r^{-1}\hbar$$
$$\sigma_{\vec{x}} = 0.970r \qquad \sigma_{\vec{p}} = 3.06r^{-1}\hbar$$
$$P(\text{in sphere}) = 0.578$$

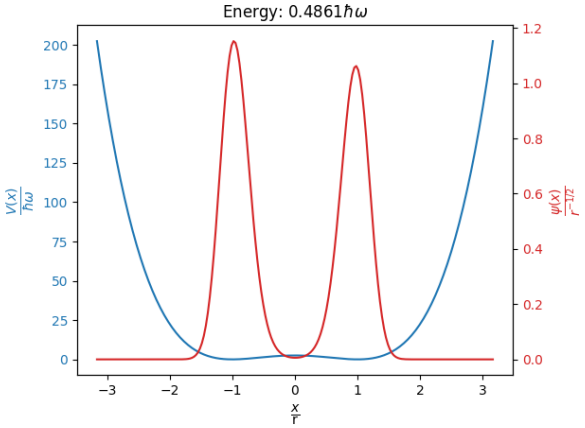

Figure 2: 1D Potential: $D = 1$, $N = 201$, $\mu = 20$, $\epsilon^2 = 0.001$

For the 2D case the observables are

$$\mu_{\vec{x}} = (3.64 \cdot 10^{-07}, -4.89 \cdot 10^{-04})^T r \qquad \mu_{\vec{p}} = (0.0, 0.0)^T r^{-1}\hbar$$
$$\sigma_{\vec{x}} = (0.687, 0.687)^T r^{-1}\hbar \qquad \sigma_{\vec{p}} = (2.13, 2.13)^T r^{-1}\hbar$$
$$P(\text{in sphere}) = 0.593$$

All the above values where calculated using a tolerance on 10^{-4} for both the conjugate gradient algorithm as well as the power method. We will look at how each parameter influences the results of the calculations.

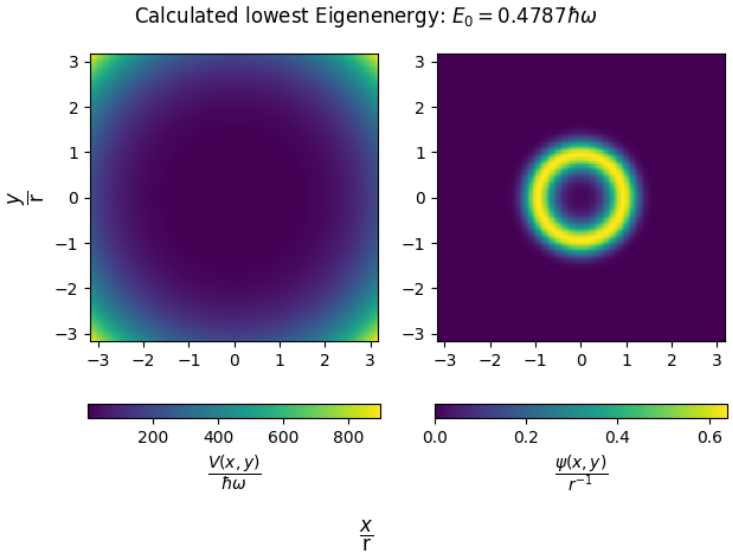

Figure 3: On the left the 2d potential, on the right the ground-state: $D = 2$, $N = 201$, $\mu = 20$ $\epsilon^2 = 0.001$

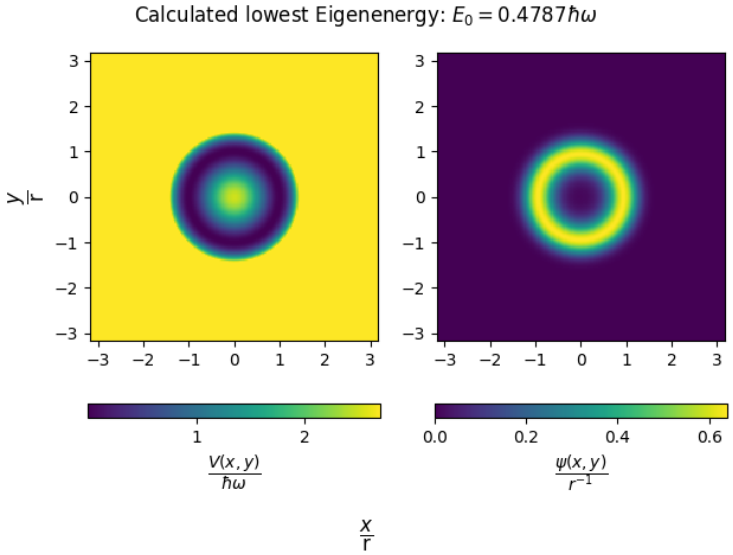

Figure 4: On the left the 2d potential, on the right the ground-state (zoomed)

5.2 Symmetric Wavefunction

As the ground state of a symmetric potential will be symmetric as well, we can initialize the random wavefunction for the power method to be symmetric as well. This might also help speeding up the algorithm as the component of the ground state in the starting vector is bigger to begin with.

5.3 Tolerance of the Conjugate Gradient Algorithm

We first looked at how the tolerance of the conjugate gradient algorithm influences the result of the calculations. We kept all parameters fixed at the script values. The tolerance for the power method was also kept at 10^{-4}. There was no visible effect on the wavefunction itself when plotting. The ultimate effect of the tolerance of the conjugate gradient seems to be quite low. However as it is computationally not that expensive to push the tolerance, we can set it quite low.

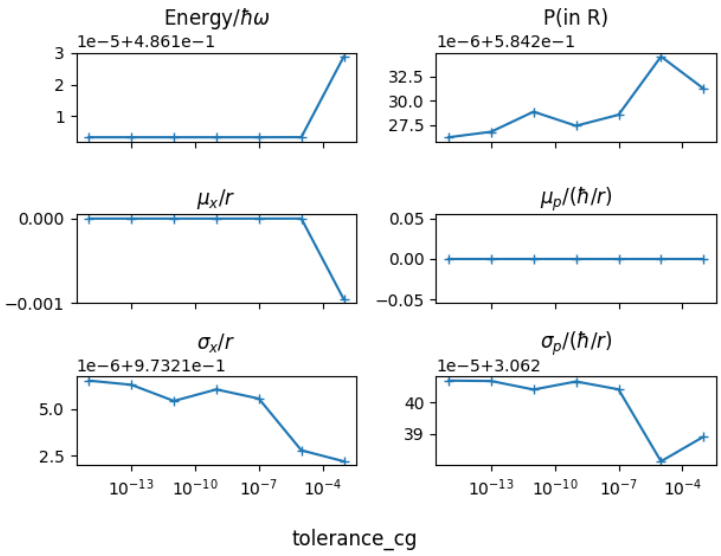

Figure 5: Evolution of observables for lower and lower tolerances.

5.4 Tolerance of the Power method

This time we kept the tolerance of the conjugate gradient algorithm fixed at 10^{-15} and varied the tolerance of the power method.

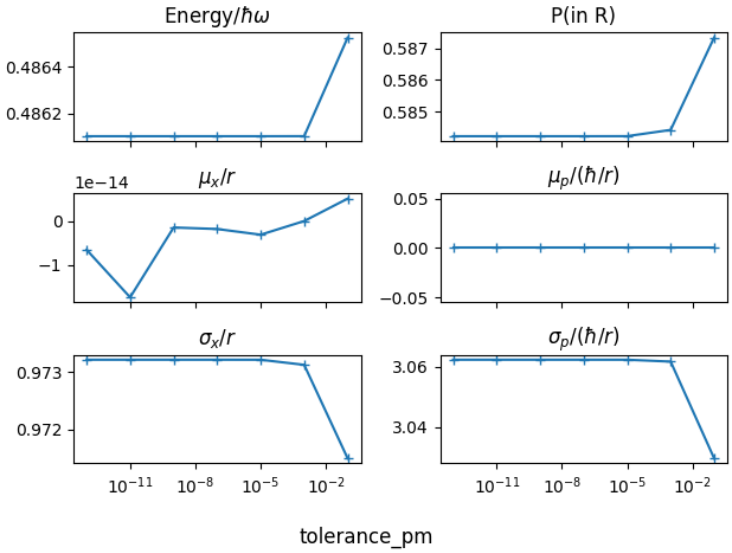

Figure 6: Evolution of observables for lower and lower tolerances in the power method.

The effect of the tolerance for the power method also seems quite low. However the effect is also diminished through the fact that we used a symmetric wavefunction to begin with. Otherwise a $|\mu_x|$ would also decrease for lower tolerances, where as in our tests in is essentially 0 to start with. For the next tests we set the conjugate gradient tolerance to 10^{-10} and the power method tolerance to 10^{-7}. For lower tolerances we ran into problems with unreasonably high computation time, especially for higher values of N.

5.5 Continuum Limit

If we want to have a finer grid for the same regime, we must send $\epsilon \to 0$ and $N \to \infty$ while keeping ϵN fixed. For the 1D case we did this for (2N+1)-values from 21 to 2001.(ϵ was tuned accordingly). The results can be seen in 7.

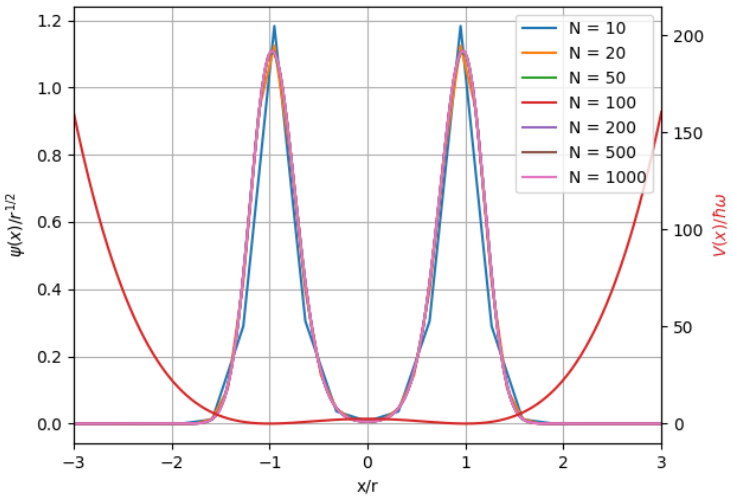

Figure 7: Continuum Limit for different values of N and ϵ while keeping $N\epsilon$ constant

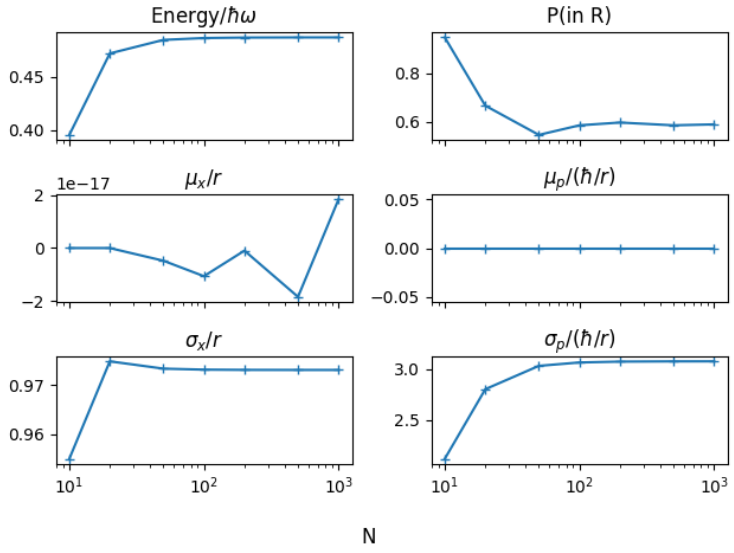

Figure 8: Evolution of observables for finer and finer grid.

5.6 Infinite Volume Limit

The infinite Volume limit was calculated in a similar way as the continuum limit. This time however, we keep ϵ fixed (we take $\epsilon = 0.00317$, the smallest value from the continuum analysis) and increase N. N was varied over a range from 2N+1=401 to 20001.

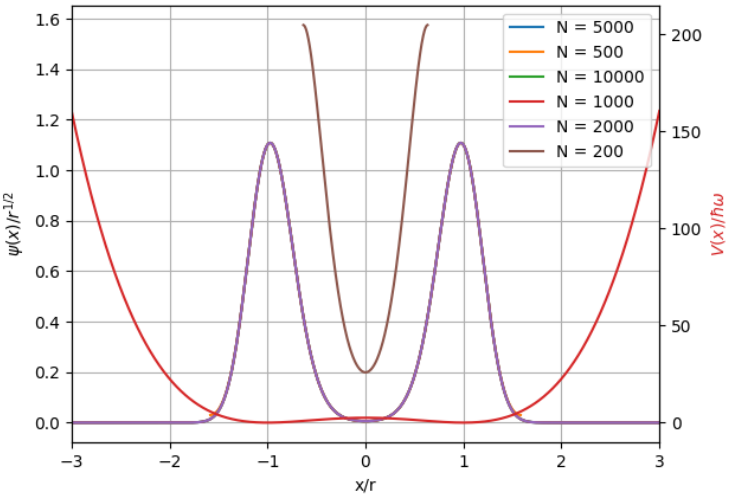

Figure 9: Infinite Volume Limit for different values of N

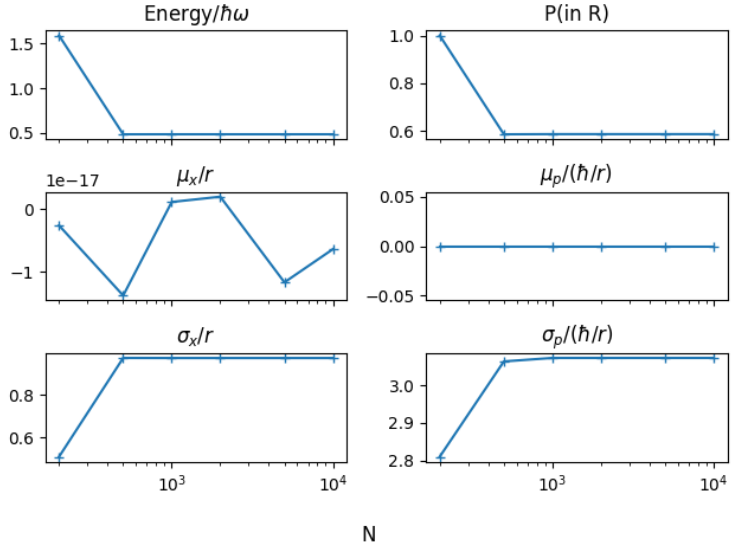

Figure 10: Evolution of observables for a bigger and bigger grid.

5.7 Remarks

5.7.1 Wave-function

- We observe two peaks of the wave function, which are localized in the local minima of the potential, as expected.
- There is an asymmetry of the wave-function which is unexpected for a symmetric Hamiltonian. We observed different heights of the peaks for different executions of the same program, which made us conclude that the asymmetry is caused by a computational error. A possible way out to minimize the error is allowing lower tolerance values in the functions that we used. We also showed that initializing a symmetric wavefunction in the first place will have a big influence on this effect. This effect is visible in the expectation value in $\vec{\mathbf{x}}$. For the 2D case this error is smaller, which is not necessarily trivial to expect.
- **Continuum limit**: The imbalance in the peak-height seems to reduce for a smaller lattice spacing. For $L = 6r$, all observables are relatively stable $2N + 1 > 100$.
- **Infinite Volume limit** For a constant lattice spacing, the wavefunction looks qualitatively the same, expect when $L < r$, which does make sense, concidering the wavefunction *can not see* the high potential barrier. The observables are mostly also stable for $L > 3r$.
- **The Heisenberg uncertainty principle** is true for both 1D& 2D, however $\sigma_x \sigma_y$ is not approximately $\frac{\hbar}{2}$, like in the QHO case.

5.7.2 Energy

- For the energy, we make the following Taylor approximation around minima (x=a) for our potential (first derivative does not appear because we are at minima):

  $$V(x) = V(a) + \frac{1}{2} V\text{“}(a)x^2 + ...$$

  With

  $$\partial_x V = \frac{mw^2}{2r^2}(x^2 - r^2)x$$
  $$\partial_x^2 V = \frac{mw^2}{2r^2}(x^2 - r^2) + \frac{mw^2 x^2}{r^2}$$

  We see (with the help of the plots too) $x = \pm r$ are the minima. If we plug $x = \pm r$ to the Taylor approximation:

  $$V = \frac{mw^2 x^2}{2} + ...$$

  So in the quadratic order, the problem is an harmonic oscillator! This means that we must get energy values close to $0.5\hbar w$, which is the ground state of harmonic oscillator. This is what we also observed in our results!
- **Continuum limit**: The energy value does not drastically change for the finite case, which is a good sign for convergence of the code. Moreover, the calculated energies for continuum and infinite volume limit cases are the similar, meaning just taking the $a \to 0$ and $L \to \infty$ limit does not affect the result significantly after $L \approx 3r$.