Department of Physics
Umeå Universitet March 22, 2018

Wave Packet Dynamics

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

March 22, 2018

Abstract

According to the Heisenberg principle, it is impossible to know the exakt position and and momentum of a quantum particle at the same time. This motivates the use of a wavefunction describing the probability of a particle at a given position. When simulating the dynamics of such particle, that is by solving the time dependent Schrödinger equation, one must know the wavefunction all over space at each time step. This makes the problem extremely difficult and computationally heavy. In this report, a quantum particle is simulated using two different numerical methods, the split operator method and the second-order differencing method. The particle is simulated as a free particle, hitting a potential well and a potential barrier. For the well and the barrier, the reflection and transmission coefficient was computed and compared to analytical expressions. The results of this was in good accordance to what was expected. The numerical solutions seemed to undershoot the transmission coefficient near the resonating energies, that is for energies close to the eigenenergies of the potential well. This could be explained as the simulated particles contained a distribution of wavenumbers and the analytical expressions was basen on monochromatic waves. The two numerical methods was then compared with respect to computation time and the results showed that the split operator method was faster. There is however room for improvements on the code for the second-order differencing method which is why more testing must be done before one can conclude this for sure.

Numerical Methods in Physics
Supervisor:Claude Dion

Contents

1 Introduction

Quantum particles does not behave like ordinary particles. This is due to the fact that their motion cannot simply be determined by the position and velocity of the particle, that is, because their position is not well defined. According to the Heisenberg principle, it is impossible to determine the exact position and momentum of a particle at the same time. The position can however be modeled by a probability density, spanning all over space. This makes it possible only to find analytic solutions to the dynamics of a quantum particle for only a few sets of potentials and initial conditions. This is why numerical methods are of high importance when studiyng quantum effects. It is possible to model and simulate these by solving the time dependent Schrödinger equation (TDSE) numerically using high level algorithms. The complexity of this situation is that since the position is described by a probability density, a wavefunction, one must store and manipulate data from all over space in each time step in comparison to ordinary particles where only the position an momentum is needed. This makes the problem of solving TDSE non-trivial and only high performance computers are able to finish the task within reasonable time.

In this report, TDSE will be solved for a free particle, a particle hitting a potential well and a potential barrier and the quantum effects that occur will be studied. The reflection and transmission coefficients for these cases will be computed and compared to analytical expressions and conclusions will be drawn for their validity. Two different algorithms will be implemented for solving TDSE and at the end of the report, a short study comparing their performance in means of computational time will be done.

First follows some theory about quantum mechanincs with and emphasis on how TDSE can be solved using two different approaches. Then some analytical results will be presented followed by how the algorithms has been implemented. Then follows the numerical results and analysis.

2 Theory

The dynamics of a quantum particle is described by the time dependent Schrödinger equation

$$i\hbar\frac{\mathrm{d}}{\mathrm{d}t}\psi(x,t) = \hat{H}\psi(x,t) \tag{TDSE}$$

where $\hbar$ is planck's constant, ψ is the wavefunction of the particle, $i = \sqrt{-1}$ and $\hat{H}$ is the Hamiltonian

$$\hat{H} = \hat{T} + \hat{V} = -\frac{\hbar^2}{2m}\frac{\mathrm{d}^2}{\mathrm{d}x^2} + V(x), \tag{1}$$

the sum of the kinetic and potential operator. In this equation, m is the mass of the particle and $V(x)$ is the potential. The naive solution to Eq. (TDSE) is

$$\psi(x,t+\Delta t) \approx e^{-\frac{i\Delta t}{\hbar}\hat{H}}\psi(x,t) \tag{2}$$

for any small step size Δt. The problem with this naive solution is that taking the exponential of an operator is non-trivial.

2.1 Split operator method

One way to get around this is by noting that the kinetic operator includes a second derivative in position space but a a simple multiplication in momentum space. Namely that

$$e^{\frac{i\Delta t\hbar}{2m}\frac{\mathrm{d}^2}{\mathrm{d}x^2}}\psi(x,t) \longleftrightarrow e^{-\frac{i\Delta t}{2m\hbar}p^2}\psi(p,t),$$

where p is the momentum of the particle. This motivates the splitting of the hamilton operator into two parts, taking the potential operator in position space and the kinetic operator in ordinary space. The final time stepping can thus be written as

$$\psi(x,t+\Delta t) = e^{-\frac{i\Delta t}{2\hbar}\hat{V}}\mathcal{F}^{-1}e^{-\frac{i\Delta t}{\hbar}\hat{T}_p}\mathcal{F}e^{-\frac{i\Delta t}{2\hbar}\hat{V}}\psi(x,t) \tag{3}$$

where $\hat{T}_p$ denotes the kinetic operator in momentum space and $\mathcal{F}$ and $\mathcal{F}^{-1}$ denotes the direct and inverse fourier transform to go from position space to momentum space and vice versa. Note that the potential operator is also split into two parts in order to make the operation unitary and thus reversible.

2.2 Second-order differencing

Another approach to this problem is to make a taylor expansion of the exponential. By doing this in one direction and again in the other and taking the difference yields the Second-order differencing method. Namely by subtracting

$$\begin{aligned}\psi(t+\Delta t) &= e^{-\frac{i\Delta t}{\hbar}\hat{H}}\psi(t)\\ \psi(t-\Delta t) &= e^{\frac{i\Delta t}{\hbar}\hat{H}}\psi(t)\end{aligned} \tag{4}$$

to get

$$\psi(t+\Delta t)-\psi(t-\Delta t) = (e^{-\frac{i\Delta t}{\hbar}\hat{H}} - e^{\frac{i\Delta t}{\hbar}\hat{H}})\psi(t). \tag{5}$$

By expanding the exponentials we get

$$\begin{aligned}e^{-\frac{i\Delta t}{\hbar}\hat{H}} &= 1-\frac{i\Delta t}{\hbar}\hat{H}-\frac{\Delta t^2}{2\hbar^2}\hat{H}^2+\frac{i\Delta t^3}{3!\hbar^3}\hat{H}^3+\mathcal{O}(\Delta t^4)\\ e^{\frac{i\Delta t}{\hbar}\hat{H}} &= 1+\frac{i\Delta t}{\hbar}\hat{H}-\frac{\Delta t^2}{2\hbar^2}\hat{H}^2-\frac{i\Delta t^3}{3!\hbar^3}\hat{H}^3+\mathcal{O}(\Delta t^4)\end{aligned} \tag{6}$$

which leads to

$$(e^{-\frac{i\Delta t}{\hbar}\hat{H}} - e^{\frac{i\Delta t}{\hbar}\hat{H}})\psi(t) = -\frac{2i\Delta t}{\hbar}\hat{H}+\mathcal{O}(\Delta t^3). \tag{7}$$

By neglecting the higher order terms the updating scheme for this method can be written as

$$\psi(t+\Delta t) = -\frac{2i\Delta t}{\hbar}\hat{H}\psi(t)+\psi(t-\Delta t). \tag{8}$$

The main difficulty with this method is that two previous values are needed in order to start the sequence. A common solution to this is to take one Euler step back in time in order to find ψ_{-1}. More about how this was implemented is found in the method section, Sec. 3.2.

2.3 The free particle

In order to study the numerical algorithms it is good to first try them on something with a known solution. Consider the initial gaussian wavefunction with central wavenumber k_0 and width $2\sigma_0^2$

$$\psi(x,0) = \left(\frac{1}{\pi\sigma_0^2}\right)^{1/4} e^{ik_0x} e^{-(x-x_0)^2/2\sigma_0^2}. \tag{9}$$

The analytical solution to Eq. (TDSE) for this initial condition using $V(x) = 0$ is

$$\psi(x,t) = \left(\frac{\sigma_0^2}{\pi}\right)^{1/4} \frac{e^{i\phi}}{(\sigma_0^2 + it)^{1/2}} e^{ik_0x} exp\left[-\frac{(x - x_0 - k_0t)^2}{2\sigma_0^2 + 2it}\right] \tag{10}$$

where $\phi \equiv -\theta - k_0^2t/2$ and $\tan\theta = t/\sigma_0$[1]. By comparing this analytial solution to the one obtained by the numerical solution one can determine the validity and and stability of the numerical algorithm.

2.4 The potential well

Changing the potential to

$$V(x) = \begin{cases} -V_0 & , |x| < a \\ 0 & , \text{else} \end{cases} \tag{11}$$

yields a more interesting case, with V_0 being a positive constant. Outside of the well the wave will propagate as a free particle (Eq. (10)) but when it hits the well it will either transmit through or reflect back. Since the wavefunction corresponds to a probability it can be shown that the probability of transmittance and reflectance will depend on the energy of the wave, and hence depend on the central wavenumber k_0. Namely that

$$T = \frac{1}{\left(1 + [V_0^2/4E(E+V_0)]\sin^2(2a\sqrt{2m(E+V_0)/\hbar^2})\right)}, \tag{12}$$

where $E = \hbar^2k_0^2/2m$ is the average energy of the particle[1]. Now T is the probability of the particle being transmitted through the well and similarly $R = 1-T$ is the probability of reflectance. Looking at Eq. 12 on can see that it will resonate with the potential well and have maximum when

$$2a\sqrt{\frac{2m(E+V_0)}{\hbar^2}} = n\pi, \quad n = 1, 2, ... \tag{13}$$

This could be viewed as standing waves in the potential well, making it transparent to the wave packet.

2.5 The potential barrier

Changing the sign of V_0 in Eq. (11) yields a potential barrier. Now the intuitive solution would be that the the quantum particle would bounce on the barrier but in quantum mechanics, there is always a probability of transmittance, regardless of the energy of the particle. This is called tunneling. If the energy of the particle is less than the energy of the barrier, the transmission coefficient will be the same as Eq. (12) but with a change of sign from V_0 to $-V_0$. If the energy is larger than the barrier height, the transmission coefficient will be[1]

$$T = \frac{1}{\Big(1 + [V_0^2/4E(V_0 - E)]\sinh^2(2a\sqrt{2m(V_0 - E)/\hbar^2})\Big)}. \tag{14}$$

The wavenumber corresponding to the energy of the potential barrier is $k_V = \sqrt{2mV_0/\hbar^2}$.

3 Method

In order to study the quantum particle using the methods in Sec. 2 the program called `wavepacket` was used [2]. First, the initial wavefunction was implemented together with a zero potential and the wave was simulated for $6.8 \cdot 10^{-3}$s using $k_0 = 200$ and the resulting wave was analyzed. Next, the potential was replaced with a potential well. For this case simulations was done for different inital wavenumbers from 250 to 350 in steps of 10 in order to find a relation of the reflection and transmission coefficients and thus be able to verify Eq. (12). The reflection and transmission coefficients was computed as

$$\begin{aligned}
R &= \int_{-\infty}^{-a} |\psi(x,t)|^2 \mathrm{d}x \approx 0.5\psi(x=-8,t) + \sum_{i=1}^{x(i)<-a} \psi(x=-8+i\Delta x,t) + 0.5\psi(x=-a,t) \\
T &= \int_{a}^{\infty} |\psi(x,t)|^2 \mathrm{d}x \approx 0.5\psi(x=a,t) + \sum_{x(i)>a}^{x(i)<8} \psi(x=a+i\Delta x,t) + 0.5\psi(x=8,t)
\end{aligned} \tag{15}$$

The final coefficients of each simulation was then used to plot R, T against the wavenumber.

The same study was then done with the potential changed to a barrier instead of a well.

As a final study the numerical algorithm was changed and the program `diffScheme` was used. The same results as with the potential barrier was produced and the two algorithms was compared by the means of excecution time. In order to make a fair comparison both programs produced the same output and was set to the same simulated time in means of seconds.

3.1 wavepacket.c

The program `wavepacket` is a simulation program for quantum particles using the split-operator method described in Sec. (2.1). The program was used with different potentials yielding different executable files. There three potentials can be compiled using either of the commands

- `make freepart`
- `make potwell`
- `make barrier`

for the free particle, potential well and potential barrier respectively. The program uses `FFTW` in order to compute the fourier transforms so this library must be installed in order to compile the program correctly.

The simulations are then executed by adding an argument file containing the parameters (preferrably the template file in appendix A) and an argument with the initial wavenumber. If the input file is found in a folder called `input_files` then a simulation for the potential well using $k_0 = 200$ could be excetuted using the line

```
./potwell input_files/pot_well_template.in 200
```

3.2 waveDiff.c

The program implementing the second-order differencing scheme is called `waveDiff` and can be compiled by the command `make diffScheme`. No external libraries was used in the implementation of this code. The same datatypes and general funcitons (e.g. the routine for reading parameters and printing results) as in `wavepacket` was used in order to be able to compare the computation times. A similar command can be used to execute a simulation using this program. Note however that the time step used must be about $5 \cdot 10^{-8}$ in order to achieve convergence.

3.2.1 Implementation

In order to implement the update scheme as described in Sec. 2.2 we must first construct the hamiltonian. This is done by looking at the matrices corresponding to the kinetic operator and the potential operator. In order to construct the kinetic operator matrix, containing a second derivative one can make use of the central difference approximation

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2} f(x_i) \approx \frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta x^2} \tag{16}$$

to obtain

$$\hat{T} = -\frac{\hbar^2}{2m\Delta x^2} \begin{bmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & & \\ 0 & 1 & -2 & 1 & \vdots \\ \vdots & & \ddots & \ddots & \\ & & & & -2 \end{bmatrix}. \tag{17}$$

The potential operator is in this case a simple diagnal matrix

$$\hat{V} = \begin{bmatrix} V_1 & 0 & \dots & 0 \\ 0 & V_2 & 0 & \vdots \\ \vdots & & \ddots & \\ 0 & & & V_N \end{bmatrix} \tag{18}$$

This means that the diagonal elements of $\hat{H}$ and off diagonal elements are found as

$$\begin{aligned} H_{i,i} &= \frac{\hbar^2}{m\Delta x^2} + V_i \\ H_{i,i+1} = H_{i,i-1} &= -\frac{\hbar^2}{2m\Delta x^2}. \end{aligned} \tag{19}$$

The implementation of this makes use of the fact that the matrix is tridiagonal and computes the matrix-vector product elementwise. It also includes the prefactor of $\Delta t/\hbar$ from Eq. 7.

The function for computing this matrix-vector product is called `hamiltonian_operator` and is found in the appendix A. It is used by the following header

```
void hamiltonian_operator(const parameters params,
 double complex *psi, double complex *Hpsi);
```

where `params` is a struct as described in [2], `psi` is the wavefunction to be premultiplied and `Hpsi` is the resulting vector ofter the multiplication.

In order to get good initial conditions for the algorithm, half an euler step back in time and half step forward in time is done to get a smaller error and symmetry also in the first step. Namely by computing

$$\begin{aligned} \psi_{-1/2} &= \left(1 + \frac{i\Delta t}{2\hbar}\hat{H}\right)\psi_0 \\ \psi_{1/2} &= \left(1 - \frac{i\Delta t}{2\hbar}\hat{H}\right)\psi_0 \end{aligned} \tag{20}$$

which changes Eq. (7) to

$$\psi_{n+3/2} = -2\frac{i\Delta t}{\hbar}\hat{H}\psi_{n+1/2} + \psi_{n-1/2}. \tag{21}$$

In order to perform this update, two functions are needed;

- `void take_step(double complex type,double complex *Hpsi, double complex *psi,parameters params);`
- `void swap_vectors(double complex *psi1,double complex *psi2,parameters params);`

The first performs the actual step and the second swaps the elements of two vectors. This is needed in order to prepare for the next step to be taken. The function `take_step` takes an argument `type` which is the type of step to be taken. This is a purely complex value including the last prefactor in Eq. (20)-(21). That is, one of the following $\{0.5i, -0.5i, -2i\}$. The function itself performs the following update

$$\psi \leftarrow \psi + \text{type} \cdot H\psi$$

which is why the swapping function is needed in order do perform the uptade scheme in Eq. (21).

In addition to these functions the routine for computing the reflection and transmission coefficient is found in the function `user_observe` as described in [2] and implements Eq. (15) in a straightforward manner.

4 Results

As a first test, a wavefunction starting at $x_0 = -2$ with width $\sigma_0 = 0.1$ and central wavenumber $k_0 = 200$ was simulated for $6.8 \cdot 10^{-3}$s using a zero-potential. The resulting wavefunction is found in Fig. 1. In the figure, the analytical result from Eq. (10) is also plotted in order to see any discrepancies in the result. As seen in the figure, the numerical result correspond well to the analytical and thus, the numerical model seem valid. In order to further prove this, the imaginary part of the numerical and analytical wavefunction is plotted in Fig. 2. The solutions match well also in this figure, and further analysis of other potentials are now possible.

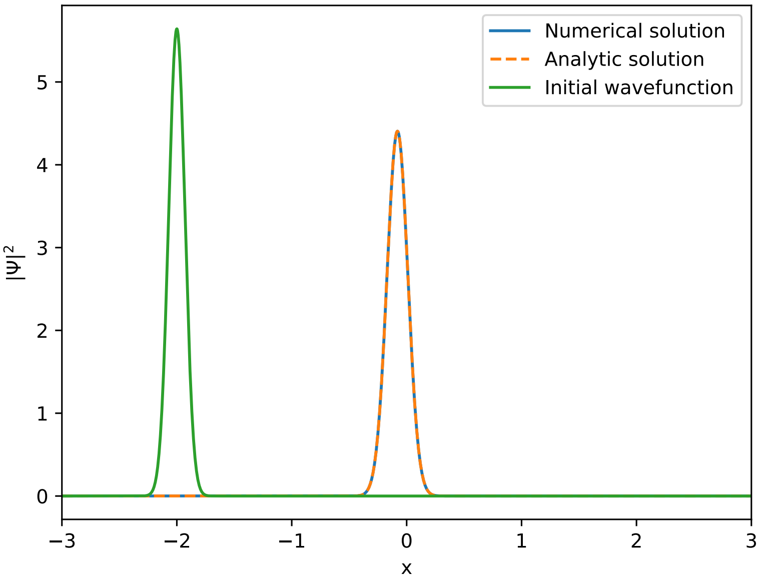


Figure 1: *Numerical and analytic solution of the time dependent schrödinger equation for a free particle.*

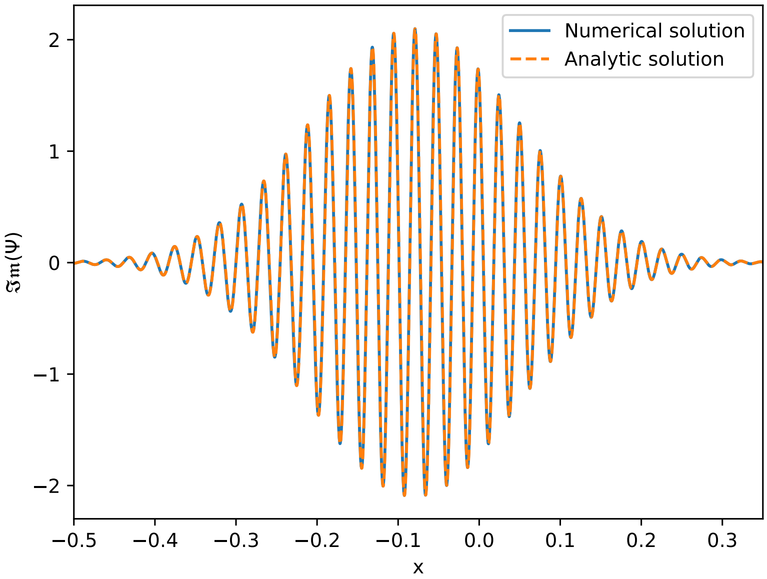
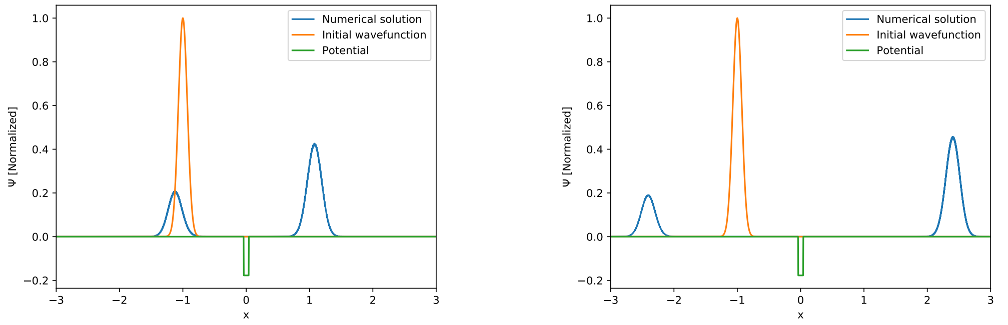


Figure 2: *Imaginary part of the solution in Fig. 1.*

The potential was changed to the potential well in Eq. (11) with depth $V_0 = 10^5$ and width $a = 0.04$. The wave started with central wavenumber $k_0 = 170$ and 280 and the resulting wavefunctions are plotted in Fig. 3. In the figure, the waves are normalized to the initial wavefunction and the potential is in the figure but not to scale. For these waves, the reflection and transmission coefficients was computed and is found in Fig. 4. The reflection coefficient starts at unity and decreases when the wave hits the well and then settles on a value which is taken as the true reflection coefficient. The same applies for the transmission coefficient. It is interesting to see in Fig. 3 that the wave in split into two, where one has passed through the well and one is reflected. It is thus a matter of probability of where the particle is after hitting the well. This means that if several particles is sent through a well like this, a percentage of them will be reflected and the rest will be transmitted. The result is thus not to be interpreted as the particle is actually being split into two rather than it is more probable that the particle will pass through. Noting from Fig. 4 that the coefficients are different for different initial wavenumber leads us to the next analysis.

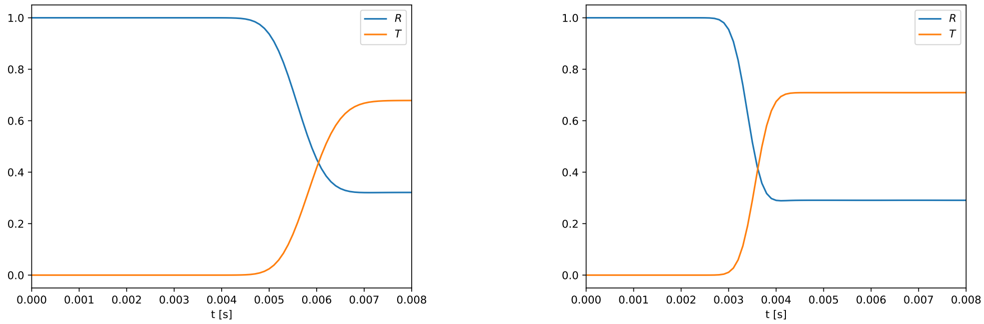
Running simulations for particles with wavenumbers from $k_0 = 150$ to $k_0 = 350$ in steps of 10, collecting the resulting reflection and transmission coefficients and plotting them together with the analytic expression Eq. (12) yields Fig. 5. The numerical values has been interpolated using cubic splines in order to see the bahaviour more clearly. In the figure, the resonating wavenumbers for Eq. (13) has been added. The numerical values seem to have the same behaviour as the analytical but tend to undershoot the peaks of the analytical expression. The reason for this is that the analytical expression is based on a monochromatic wave consisting of only one wavenumber. This means that the wave will be resonating with the well perfectly for some wavenumbers and the transmission will be equal to one. Having the same central wavenumber in the numerical solution, only a fraction of the wavenumbers will be in perfect resonance with the well meaning that also a fraction will be reflected. This means that even though the numerical result does not match the analytical results perfeclty, it does not mean that the result is bad, only that the circumstances are different. It is also interesting to see in Fig. 5 that the reflection coefficient tend to decrease with larger wavenumber. This makes sense since the wavenumber ralates to the energy of the particle, and with higher energy, the wave should more easily transmitt through the potential well.



(a) $k_0 = 170$

(b) $k_0 = 280$

Figure 3: *Numerical solution of the time dependent schrödinger equation of the potential well. The potential in the figure is not to scale.*



(a) $k_0 = 170$

(b) $k_0 = 280$

Figure 4: *Reflection and transmission coefficient of the particles in Fig. 3.*

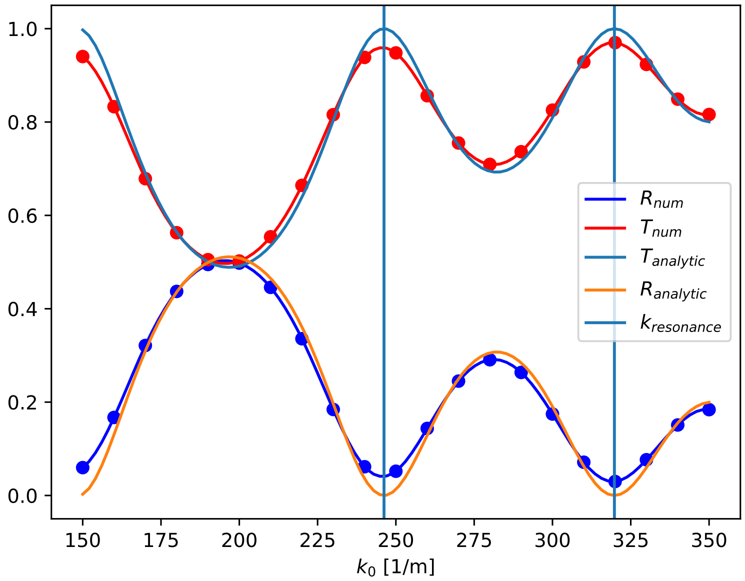
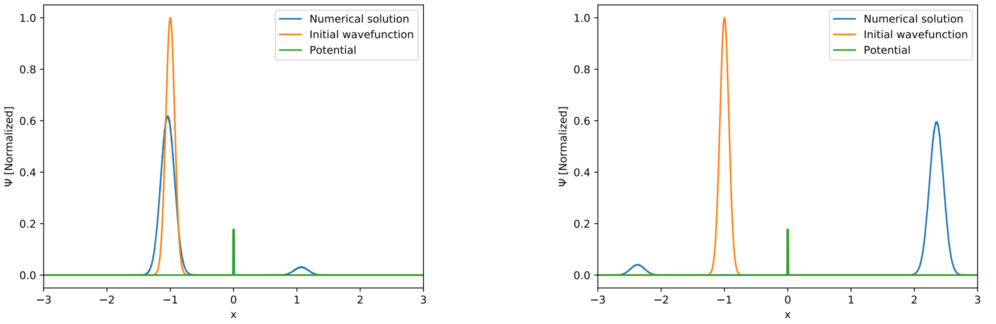


Figure 5: *Reflection and transmission coefficient for different center wave numbers used.*

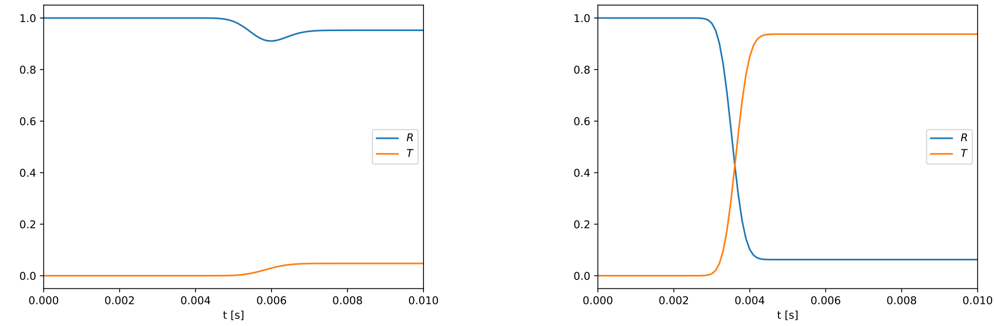
In the next case, the potential barrier was replaced with a potential barrier of height $V_0 = 2 \cdot 10^4$ and witdh $a = 0.01$. The same waves as in the previous test was simulated and is found together with the potential and the initial wavefunction in Fig. 6. In the figure, it seem like the wave is mostly reflected for the low wavenumber of $k_0 = 170$ and mostly transmitted for the high wavenumber case $k_0 = 280$. This can also be seen in Fig. 7 where the reflection and transmission coefficients are plotted against time. In Fig. 7a the reflection coefficient has a slight dip when the wave is hitting the barrier. This might be due to numerical errors as the wave is ”inside” the barrier. As with the potential well only the final values from these figures are important as it will reflect the probability of reflection of many particles hitting the barrier. The wavenumber corresponding to the height of the barrier is in this case $k_V = 200$ which makes Fig. 7a even more interesting. This tells us that even though the particle has less energy than what would be required to classically pass the barrier, there is a slight probability of getting through. This effect is known as tunneling and the result proves this remarkable quantum theory.

Doing the same sweep of simulations, starting from $k_0 = 150$ to $k_0 = 350$ in steps of ten gives Fig. 8. The analytical expression for the transmission coefficient Eq. 14 is included in the figure. The wavenumber corresponding to the barrier height is marked as a line in the figure. In this case is seem like the numerical solution correspond even better to the analytical expression the for the potential well. The same phenomena occur however that the numerical result is undershooting the first extreme. It is interesting to see that close to k_V the transmission coefficient increases rapidly which makes sense.



(a) $k_0 = 170$ **(b)** $k_0 = 280$

Figure 6: *Numerical solution of the time dependent schrödinger equation of the potential barrier. The potential in the figure is not to scale.*



(a) $k_0 = 170$ **(b)** $k_0 = 280$

Figure 7: *Reflection and transmission coefficient of the particles in Fig. 6.*

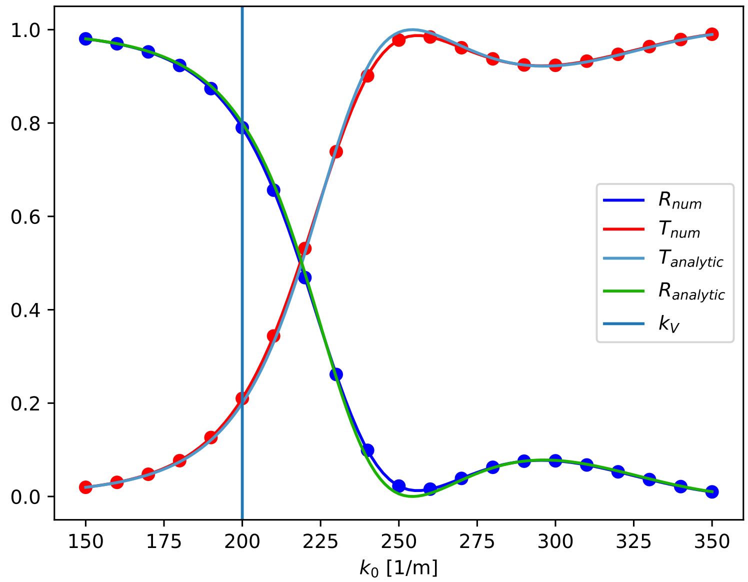
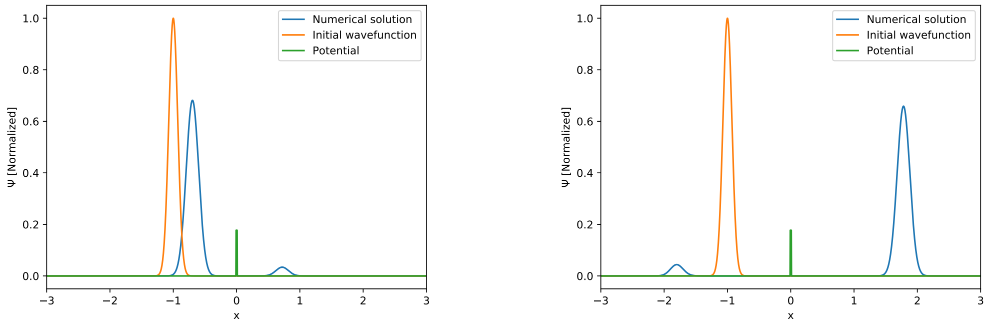


Figure 8: *Reflection and transmission coefficient for different center wavenumbers used. k_V is the wavenumber corresponding to the potential barrier height.*

As a final experiment the second-order differencing scheme was implemented and all results from the potential barrier was reproduced. As seen in Fig. 9-11 the results are practically identical to those of the split operator method. In order to see that there actually is a difference the numerical solutions in Fig. 8 and Fig. 11 was subtracted and plotted in Fig. 12. This discrepancy in results is too small for the eye to see in one of the original figures meaning that their results can be taken as equal.

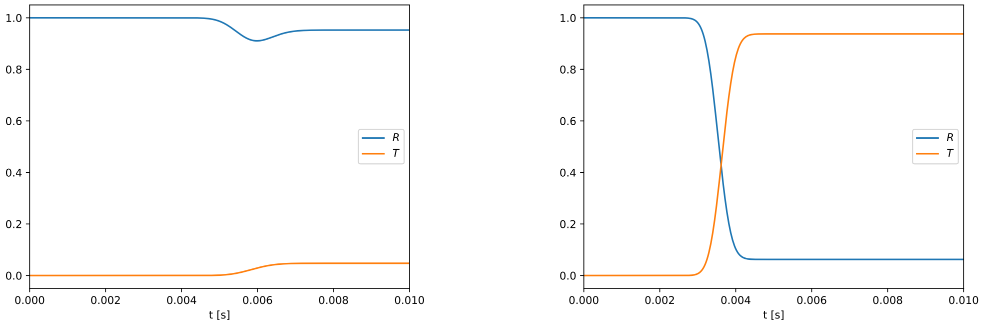
Running a simulation using both methods for with the same IC and simulation time gives a way of comparing the computation times of the algorithms. The second-order differencing scheme needed a shorter time step $\Delta = 5 \cdot 10^{-8}$ in order to converge and thus more steps $N_t = 245000$ had to be taken. The split method had $\Delta t = 2 \cdot 10^{-7}$ and took $N_t = 60000$ steps. The time for these simulations was 215.30s and 151.04s respectively. This means that the split operator method is in fact faster that the other method and this is probably the reason why `wavepacket.c` uses this. One thing one must note is that even though a lot of the same code was used in the second-order differencing code in order to make a fair comparison this might not be the best way to do it. There may be faster implementations to this and probably a lot of code optimization to be done before it works optimally. The split operator method code is old in comparison and a lot of optimization of it has been done in order for it to perform at its fullest. This means that even though the results here show that this method is faster, more tests and optimization must be done for the second-order differencing code before any hard conclusions can be drawn about the computation times.



(a) $k_0 = 170$

(b) $k_0 = 280$

Figure 9: *Numerical solution of the time dependent schrödinger equation of the potential barrier using the second-order differencing scheme. The potential in the figure is not to scale.*



(a) $k_0 = 170$

(b) $k_0 = 280$

Figure 10: *Reflection and transmission coefficient of the particles in Fig. 9.*

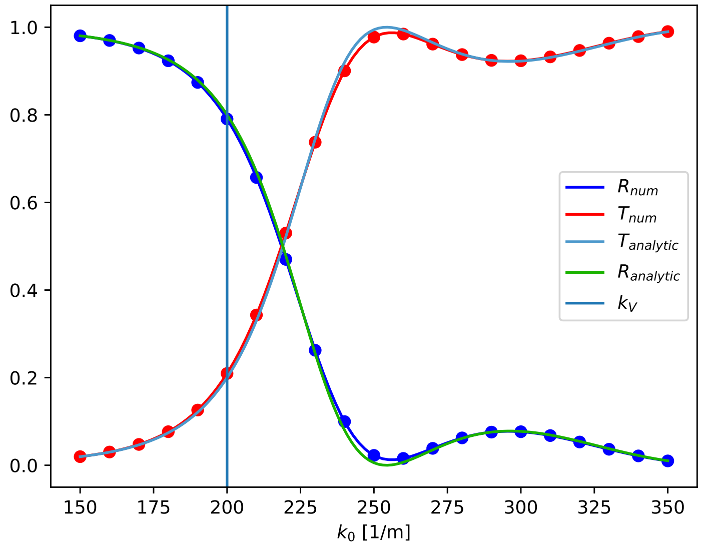


Figure 11: *Reflection and transmission coefficient for different center wavenumbers used using the second-order differencing scheme.* k_V *is the wavenumber corresponding to the potential barrier height.*

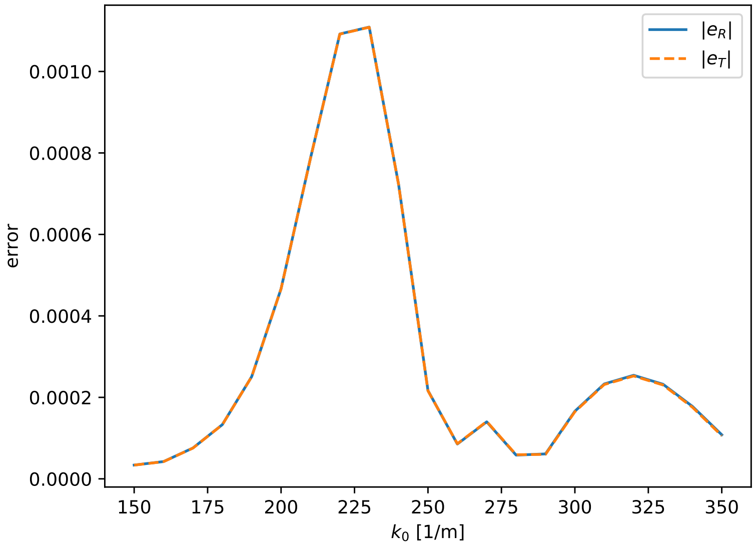


Figure 12: *The difference between the results from the split operator method and the second-order differencing scheme versus central wavenumber.*

5 Conclusion

As a conclusion one can say that the numerical results for simulating a quantum particle performs well in both predicting the dynamics of it and verifying quantum effects which are otherwise only proven analytically for simple cases and experimentally. The results showed that the wavepacket can be reflected when meeting a potential well and, by resonance, it can also be almost perfectly transmitted. The numerical transmission coefficient matched the analytical expression quite well but undershot the peak values in transmission. This is due to that the analytical expression is basen on a monochromatic wave and the simulated wave was a wavepacket consisting of waves with a finite bandwidth. Similar results was obtained for the potential barrier in which also the tunneling effect could be proven. This means that a quantum particle meeting a more energetic potetial barrier might still pass through it.

When comparing the two algorithms split operator method and second-prder differencing method the split operator method was a lot faster with respect to computational time. If this result is perfectly valid is not clear since there might be some code optimization still possible to do on the second-order differencing code. In order to be able to draw more conclusions regarding this one must first compare different datatypes and try to make the code more efficient. This report showed however that the split operator method is more likely to be faster.

References

[1] J. J. Sakurai and J. Napolitano, Modern Quantum Mechanics (Cambridge University Press, Cambridge, 2017

[2] C. M. Dion, A. Hashemloo, and G. Rahali, Program for quantum wavepacket dynamics with time-dependent potentials, Comput. Phys. Commun. 185, 407 (2014).

A Codes

code/input_files/pot_well_template.in

```
units = AU
mass = 1.
nx = 49152
x_min = -8.
x_max = 8.
dt = .0000002
nt = 60000
nprint = 500

output = norm
output = energy
output = x_avg
output = sx
output = user_defined
```

code/hamiltonian_operator.c

```c
void hamiltonian_operator(const parameters params, double complex *psi,
    double complex *Hpsi){

  double htmx2,th;
  double Hii,Hi1;
  double complex Hp1,Hp2,Hp3;
  extern double *V;
  int N = params.nx_local;

  htmx2 = params.hbar*params.dt/(params.mass*params.dx*params.dx);
  th = params.dt/params.hbar;

  Hi1 = -0.5*htmx2;
  Hii = htmx2 + th*V[0];

  Hp2 =Hii*psi[0];
  Hp3 =Hi1*psi[1];

  //Set first element
  Hpsi[0] = Hp2 + Hp3;

  for(size_t i=1;i<N-1;i++){
    Hii = htmx2 + th*V[i];

    //Compute sum elements
    Hp1 = Hi1*psi[i-1];
    Hp2 = Hii*psi[i];
    Hp3 = Hi1*psi[i+1];

    //Set element i
    Hpsi[i] = Hp1 + Hp2 + Hp3;
  }

  Hii = htmx2 + th*V[N-1];

  Hp1 = Hi1*psi[N-2];
  Hp2 = Hii*psi[N-1];

  //Set last element
  Hpsi[N-1] = Hp1 + Hp2;

}
```