JavaPsi
Simulating and Visualizing Quantum Mechanics

Marcel Schmittfull
Celtis-Gymnasium Schweinfurt
Germany

Contents

1 Introduction

The aim of this project is to simulate and visualize the behavior of particles in quantum mechanics with an interactive Java program. Almost all input data are modifiable by user interaction during runtime. In particular, the shape of the potential in which the particle 'lives' can be entered freely by mouse and the eigenfunctions are shown by selecting the corresponding energy eigenvalues.

The program computes time-dependent or time-independent solutions, either of one-, two- or three-dimensional Schrödinger equation. In the one-dimensonal time-independant case the wave function is numerically calculated by the Numerov algorithm, the energy eigenvalues are found by a shooting method. The complex-valued time-dependant wave-function, which is evaluated using the Goldberg-Schey-Schwartz algorithm, can either be animated or plotted as a function of space and time, whereby the real and imaginary parts are visualized using a color map.

The wave-functions of a three-dimensional spherical potential are calculated by reducing the problem to the one-dimensonal case using radial and spherical eigenfunctions and solving the problem again using Numerov's algorithm.

Comparisons of the calculated wave-functions with analytical solutions of well-known potentials (e.g. free particle, simple boxes and walls, harmonic oscillator, etc.) are made and show the accuracy of the code.

Especially, the program is suitable for educational purposes due to its ability to elucidate a large variety of quantum mechanical principles and effects, such as wave property of particles, tunnel effects, or the behavior in arbitrary potentials.

2 Basics of quantum mechanics

Since the behavior of microscopic particles such as electrons and photons cannot be described by classical mechanics, quantum mechanics was developped in the beginnging of the 20th century. The following section shall introduce the basical physical background of this theory.

2.1 Superposition of states

One central idea of quantum mechanics is the *state* of a particle. In contrary to classical mechanics, it is not possible to measure all values of a state with arbitrary accuracy. For example, one cannot exactly determine the momentum and the position of a particle at the same time. Particles can be in a *superposition* of several states, whereby it is interesting to note that these different states of a particle may interact with each other leading for example to the interference observed in the double slit experiment. Even if only one single particle goes through the double slit, interference can be observed. Thus, the particle is in a superposition containing the ways through the two slits.

Now, if a particle is in a state S which is a superposition of the states A and B, one can observe that the result s, which results from a measuring of the observable O on the particle in state S, are *not* 'mixtures' of the results a and b, which would result from measuring O on the particle in the states A and B respectively. Instead, s is always *either* completely a *or* completely b. Hence, when measuring V on the particle in state S, this particle 'jumps' [1] in either the state A or the state B. The states A and B, or more generally all states into which the particle can jump, are called *eigenstates*. The results a and b or rather all obtainable results from a measuring of O are called *eigenvalues* of the observable O.

2.2 Indeterminism and probability interpretation

Since measuring the superposition S of A and B does not yield mixtures of a and b but either a or b, it is not possible to predict the outcome of measuring of the observable O on the particle in the state S with absolute certainty. In fact one can merely say 'sometimes a and sometimes b' with absolute certainty. Due to this fact we have to give up determinism in quantum mechanics! Wheather the obtained result z is a or b can only be said by a certain *probability*. Hence, the position of an electron in an atom for example cannot be evaluated explicitly any more. Instead one can only evaluate the *probability density* $P(x)$ for the position of the electron. Due to Born, this probability density is the absolute-square of a probability amplitude – the so-called *wave function* $\psi(x)$:

$$P(x) = |\psi(x)|^2. \tag{2.1}$$

Thus, knowing the wave function $\psi(x)$ we can calculate the probability density $P(x)$ and find the most probable position.

The probabilities $P(x) = |\psi(x)|^2$ still have to satisfy a confining condition, for observing the entire existing area $x = -\infty$ to $x = \infty$ or rather the entire allowed area $x = x_0$ to $x = x_1$ one obtains the *normalization condition* that the sum of all probabilities has to yield 1, i.e.

$$\int_{x_0}^{x_1} P(x)\mathrm{d}x = 1. \tag{2.2}$$

In order to fulfill this condition the programs described in the following sections contain a short algorithm which normalizes the wave function after its actual calculation.

2.3 Heisenberg's uncertainty principle

Heisenberg*'s uncertainty principle* says that the information about the position x of a particle has to be the more inaccurate, the more accurate the information about the momentum p of this particle is, and vice versa that the information about the momentum p must be the more inaccurate, the more accurate the information about the position x is. That is, the accuracy Δx of the information about the position is inversely proportional

to the accuracy Δp of the information about the momentum, i.e.

$$\Delta x \Delta p \geq \hbar/2 \qquad (\hbar = \frac{h}{2\pi} \approx 1,055 \cdot 10^{-34}\,\mathrm{J\ s}) \tag{2.3}$$

with $\hbar$ being an universal constant. The changeover from quantum mechanics to classical mechanics means to assume $\Delta x \Delta p = 0$. That is, quantum mechanics is a generalization of classical mechanics.

2.4 Dirac's model

As already mentioned in section 2.1 the states of a superposition interact with each other. Now P. A. M. Dirac [1] developped a very succesful and interesting mathematical framework for the description of these interactions.

A state is represented by a vector in a Hilbert room. Due to Dirac these vectors are called *kets* and denoted by for example $|S\rangle$. All properties of a state must be contained in a ket.[1] The dual vectors of the bras which can be seen as kind of opposites of the kets are called *bras* and denoted by for example $\langle S|$. Observables are represented by *linear operators*. In the special case

$$\alpha|S\rangle = a|S\rangle, \tag{2.4}$$

the linear operator α stands only for a muliplication of the ket $|S\rangle$ with a number a, i.e. α only modifies the length of the ket but not the direction. If equation (2.4) holds, a is called *eigenvalue* of the linear operator α or rather of the observable corresponding to α. In this case, $|S\rangle$ is an *eigenket* and $\langle S|$ is an *eigenbra*. The wave function corresponding to the eigenket $|S\rangle$ and the eigenbra $\langle S|$ is called *eigenfunction*. The state corresponding to $|S\rangle$ and $\langle S|$ is an eigenzustand, compare section 2.1.

Now, Dirac's model defines some operations and rules for the kets and bras which enable us to describe the interaction of several states or rather kets and bras observed in experiments very properly. For a more detailed description of Dirac's model see [1].

3 Solving Schrödinger's equation

3.1 One-dimensional, time-independent case

In order to calculate the position probability density $P(x)$ of a particle in quantum mechanics one needs to calculate the wave function $\psi(x)$. In 1926 Erwin Schrödinger proposed his famous differential equation, the Schrödinger equation. In the one-dimensional time-independent case it is

$$\underbrace{\left(-\frac{\hbar^2}{2m}\,\frac{\mathrm{d}^2}{\mathrm{d}x^2} + V(x)\right)}_{\mathrm{H}}\psi(x) = E\,\psi(x), \tag{3.1}$$

[1] Compare definition of *state* in section 2.1.

where m denotes the mass of the particle and $V(x)$ is the position-dependent-potential. The linear operator corresponding to the total energy E is denoted by H and is in analogy to classical mechanics often called HAMILTON operator.

Basically, the SCHRÖDINGER equation is merely a 'recipe' which serves to describe the procedures observed in experiments very properly. The recipe says that the total energy E is always equal to the sum of kinetic energy $\frac{p^2}{2m}$ and potential energy $V(x)$ – it is thus the quantum mechanical analogon to the law of conservation of energy in classical mechanics. Though this recipe can be guessed from laws of classical mechanics[2], it is – like the law of conservation of energy in classical mechanics – not completely deducable.

Since analytical solutions of equation (3.1) only exist for some special potentials $V(x)$, the following sections will primarily consider numerical approaches which are able to solve the SCHRÖDINGER equation with arbitrary potentials $V(x)$.

3.1.1 Numerov algorithm

The one-dimensional time-independent SCHRÖDINGER equation (3.1) can easily be solved by using the RUNGE-KUTTA-method.[3] Yet, there is a numerical method especially tuned for the SCHRÖDINGER equation – the so-called NUMEROV- or FOX-GOODWIN-algorithm, which features a very good accuracy and a very high efficiency. Thus, the program uses this NUMEROV-algorith to solve the SCHRÖDINGER equation . It can be derived as follows.

We look for a recursion or rather an iteration which evaluates beginning with two initial boundary values $\psi(x-h)$ and $\psi(x)$ all further $\psi(x+h)$ of the desired wave function ψ.[4] In order to fulfill the normalization condition (2.2) the wave function $\psi(x)$ must converge to 0 at $x \to \pm\infty$. This is made sure by setting the potential the user may input into a box with infinitely high borders. Let the boundaries of the output screen be $x = -x_r$ and $x = x_r$. Then $\psi(\pm x_r) \to 0$ because $V(\pm x_r) \to \infty$. So we obtain our first boundary value $\psi(\pm x_r) = 0$.

The second boundary value $\psi(\pm x_r \mp h)$ is arbitrary because of the normalization done at the end of evaluating the wave function. Now, in order to find the desired recursion we first transform the SCHRÖDINGER equation (3.1) to

$$\psi''(x) + F(x)\psi(x) = 0 \qquad \text{with } F(x) = \frac{2m}{\hbar^2}\Big(E - V\,(x)\Big), \tag{3.2}$$

where $\psi''(x)$ is the second spatial derivative of $\psi(x)$. Then, we approximate

[2]Such ways of 'guessing' the SCHRÖDINGER equation are contained in almost every book on quantum mechanics, for example [6] p. 15f.

[3]In an older version of the program the SCHRÖDINGER equation (3.1) was solved with the RUNGE-KUTTA-methode of 5th order. But unfortunately the accuracy was not contenting.

[4]Here h denotes not PLANCK's constant but the increment of x.

$\psi(x \pm h)$ by the Taylor series:

$$\begin{aligned}\psi(x+h) &= \psi(x) + h\psi'(x) + \frac{h^2}{2}\psi''(x) + \frac{h^3}{6}\psi'''(x) + \frac{h^4}{24}\psi''''(x) + \cdots, \\ \psi(x-h) &= \psi(x) - h\psi'(x) + \frac{h^2}{2}\psi''(x) - \frac{h^3}{6}\psi'''(x) + \frac{h^4}{24}\psi''''(x) + \cdots.\end{aligned}$$

For the sum $\psi(x+h) + \psi(x-h)$ we obtain

$$\psi(x+h) + \psi(x-h) = 2\psi(x) + h^2\psi''(x) + \frac{h^4}{12}\psi''''(x) + O(h^6)$$

and hence

$$\psi(x+h) = -\psi(x-h) + 2\psi(x) + h^2\psi''(x) + \frac{h^4}{12}\psi''''(x) + O(h^6). \tag{3.3}$$

Remarkably, the expressions with $\psi'(x)$ and $\psi'''(x)$ vanish. Since we know $\psi''(x)$ from equation (3.2), we only need an expression for $\psi''''(x)$. (3.3) leads to

$$\psi''(x) = \frac{\psi(x+h) - 2\psi(x) + \psi(x-h)}{h^2} - \frac{h^2}{12}\psi''''(x) + O(h^4). \tag{3.4}$$

Generally

$$\psi''(x) = \frac{\psi(x+h) - 2\psi(x) + \psi(x-h)}{h^2} + O(h^2). \tag{3.5}$$

Since the expression with $\psi''''(x)$ in equation (3.3) contains the factor h^4, one obtains with equation (3.5), whose error order is h^2, a total error of order h^6 what does not worsen the earlier error order. From the equation (3.2) and (3.5) follows

$$\begin{aligned}\psi''''(x) &= \frac{\mathrm{d}^2}{\mathrm{d}x^2}\Big(-F(x)\psi(x)\Big) \\ &= -\frac{F(x+h)\psi(x+h) - 2F(x)\psi(x) + F(x-h)\psi(x-h)}{h^2} + O(h^2)\end{aligned} \tag{3.6}$$

Inserting of the obtained results in equation (3.3) yields the desired iteration

$$\psi(x+h) = \frac{\psi(x)[2 - \frac{5h^2}{6}F(x)] - \psi(x-h)[1 + \frac{h^2}{12}F(x-h)]}{1 + \frac{h^2}{12}F(x+h)} \tag{3.7}$$

3.1.2 Numerical evaluation of the energy eigenvalues

So we have found an algorithm for evaluating the wave function $\psi(x)$ for an arbitrary energy E and an arbitrary[5] potential $V(x)$. Yet, the evaluated wave function $\psi(x)$ satisfies the boundary condition $\psi(\pm x_r) = 0$ usually only at one boundary, namely the boundary one used for the initial value of ψ. At the other boundary either $\psi(-x_r)$ or $\psi(x_r)$ diverges to $\pm\infty$. The wave function $\psi(x)$ satisfies the boundary condition at both boundaries only

[5] Arbitrary except for the boundaries which must be set to infinity.

then, if the chosen energy E is an energy eigenvalue E_n. The problem of finding these eigenvalues is often referred to as two point boundary value problem. The very good efficiency of the NUMEROV algorithm suggests the so-called *shooting-method*[6] for finding the energy eigenvalues E_n.

The shooting method works as follows. If we choose the initial values of *psi* at $\psi(-x_r) = 0$ and $\psi(-x_r + h) =$ arbitrary, we simply have shots on the energy E. If $\psi(x)$ diverges to $+\infty$ or $-\infty$ for big x, our shot was an error and we have to modify E. This is repeated until $\psi(x)$ converges to 0 for big x. Observing $\psi(x)$ for big x with different energies E, one recognizes that around an energy eigenvalue E_n the right boundary of $\psi(x)$ changes its sign from $+\infty$ to $-\infty$ or vice versa. This change of sign also leads to a change of the number n of null points in $\psi(x)$. Now, by counting the number n null points of $\psi(x)$[7] one can associate every trial energy value E with a certain n. Generally, a higher number n of null points indicates a higher energy E. More detailed, every eigenfunction $\psi_n(x)$ possesses exactly *one* more null point than its preceding eigenfunction $\psi_{n-1}(x)$. Thus, we first need to search[8] for two energy values E whose number n of null points differs by exactly 1. Between these two energy values lies exactly one energy eigenvalue E_n which can be arbitrarily accurately approximated by further use of the bisection method. By repeating this procedure one obtains all desired energy eigenvalues E_n.

3.1.3 Free particle

In order to check the results of the NUMEROV algorithm and the shooting method we regard the special case of a free particle since in this case the wave function and the energy eigenvalues can be obtained analytically.

The potential of a free particle is $V(x) = 0$ for all x. From equation (3.1) we obtain

$$-\frac{\hbar^2}{2m}\frac{\mathrm{d}^2}{\mathrm{d}x^2}\psi(x) = E\,\psi(x). \tag{3.8}$$

The total energy E becomes

$$E = V + E_{kin} = E_{kin} = \frac{p^2}{2m},$$

where m denotes the mass and the p the momentum of the particle. Assuming $p = \hbar k$ (k wavenumber) in all points x we obtain

$$E = \frac{\hbar^2 k^2}{2m}. \tag{3.9}$$

Thus equation (3.8) implies

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}\psi(x) + k^2\psi(x) = 0. \tag{3.10}$$

[6] See e.g. [4] S. 757ff.

[7] This procedure of counting the points where $\psi(x) = 0$ holds is implemented by counting the number of changes of sign in $\psi(x)$.

[8] E.g. by using the bisection method.

The solutions of this equation are e^{ikx} and e^{-ikx}. Unfortunately these solutions don't satisfy the normalization constrainment (2.2) because

$$|\psi(x)|^2 = \psi^*(x)\psi(x) = e^{-ikx} \cdot e^{ikx} = 1 \tag{3.11}$$

holds for all x. The probability for the particle being at the point x is equal for all x, namely 1. That is the position of the particle is completely unknown, $\Delta x \to \infty$. Then, according to HEISENBERGs uncertainty principle (2.3) the momentum p can be determined exactly, $\Delta p \to 0$. This Dies stimmt mit der in equation (3.9) gemachten Bedingung $p = \hbar k$ überein.

Whatsoever there are several possibilities to satisfy the normalization constrainment (2.2) despite equation (3.11). One such possibilitie is DIRAC's δ function[9]:

$$\begin{aligned} \int_{-\infty}^{\infty} \delta(x)\mathrm{d}x &= 1 \\ \delta(x) &= 0 \quad \text{für } x \neq 0. \end{aligned} \tag{3.12}$$

But since the δ function is no real methematical *function*[10] another solution of the problem migth be better. Since a linear combination of two solutions always yields another solution we form the linear combination

$$\psi(x) = Ae^{ikx} + Be^{-ikx} \tag{3.13}$$

in order to satisfy the normalization constraint (2.2). The coefficients A and B of this linear combination can be obtained from the boundary conditions. In order to assure the existence of such boundary conditions in the program the potential which can be modified by the user is set into a high umgebendes äußeres potentials, i.e. in a potential box with infinite high boundaries. Thus the particle is not completely free any more, but only free in a certain area (the inner area of the big potential box). Unfortunately this constraint cannot be avoided because of the mentioned problem $\Delta x \to \infty$ for a completely free particle.[11]

3.1.4 Particle in a box

Setting the free particle potential in a box with two infinitively high boundaries, i.e.

$$V(x) = \begin{cases} 0 & \text{for } 0 \leq x \leq a \\ \infty & \text{else,} \end{cases} \tag{3.14}$$

leads to two boundary conditions $\psi(0) = \psi(a) = 0$. So we are able to calculate the coefficients A and B in equation (3.13). The normalized wave function $\psi_n(x)$ becomes[12]

$$\psi_n(x) = \sqrt{\frac{2}{a}} \sin \frac{n\pi}{a} x \qquad (n = 1, 2, 3, \dots). \tag{3.15}$$

[9] See [1] p. 58ff.

[10] Usually, every x-value of a function must be associated with at most on y-value.

[11] This constraint was also necessary for the NUMEROV algorithm in section 3.1.1

[12] For details see almost every book on quantum mechanics, e.g. [2] oder [6].

As Figure 1 shows, the numerical solution obtained by the NUMEROV algorithm is very close to this analytical solution.

With equation (3.9) the total energy E becomes

$$E = E_{kin} = \frac{\hbar^2 k^2}{2m} = \frac{n^2\pi^2\hbar^2}{2ma^2} \equiv E_n. \tag{3.16}$$

Since $n = 1, 2, 3, \ldots$ we obtain only the values E_1, $4E_1$, $9E_1 \ldots$ as possible total energy E_n, i.e. the energy is *quantisized*.[13] This quantisation of the energy E can be explained by the fact that only certain values E_n satisfy the SCHRÖDINGER equation

$$\mathrm{H}|E_n\rangle = E_n|E_n\rangle$$

in such a way that the kets $|E_n\rangle$ are eigenkets.

3.1.5 Harmonic oscillator

A particle in the harmonic oscillator potential $V(x) = \frac{1}{2}kx^2$ is another important example because of its big role in molecule physics. From the SCHRÖDINGER equation

$$-\frac{\hbar^2}{2m}\,\frac{\mathrm{d}^2}{\mathrm{d}x^2}\psi(x) + \frac{1}{2}kx^2\psi(x) = E\,\psi(x) \tag{3.17}$$

one obtains with $\omega = \sqrt{\frac{k}{m}}$, $\xi = \sqrt{\frac{m\omega}{\hbar}}x$ and $n = 1, 2, 3, \ldots$ the normalized wave function $\psi_n(x)$.[14]

$$\psi_n(\xi) = \left(n!2^n\sqrt{\pi}\right)^{-1/2} H_n(\xi)\, e^{-\xi^2/2}, \tag{3.18}$$

where $H_n(\xi)$ are the Hermite polynomials n-th degree. For the energy eigenvalues E_n one obtains

$$E_n = \hbar\omega\left(n + \frac{1}{2}\right), \qquad n = 1, 2, 3 \ldots. \tag{3.19}$$

That is the E_n are equidistant. Exactly the same result shows the output of the NUMEROV algorithm and the shooting method in figure 2

Interestingly to note is that the minimum possible energy E_0 is not 0 as in classical mechanics, but $frac12\hbar\omega > 0$. (Compare figure 2.) This at the first glance peculiar result can be explained by HEISENBERG's uncertainty principle (2.3). The classical minimum energy is 0 and is located at a point x where the potential energy V as well as the kinetic energy E_{kin} are both 0. That is, at this point x the position x is known with absolute accuracy ($V(x) = \frac{1}{2}kx^2 = 0 \Rightarrow x = 0$) and at the same time the momentum is known with absolute accuracy, too ($E_{kin} = 0 \Rightarrow p = 0$). Thus $\Delta x = \Delta p = 0$ what contradicts HEISENBERG's uncertainty principle $\Delta x \Delta p \geq \hbar/2 \neq 0$. That is

[13] In contrary to this the energy of a free particle (see section 3.1.3) is *continuary*.

[14] As above, the detailed calculation can be found in nearly every book on quantum mechanics.

$E = 0$ cannot be an energy eigenwert, i.e. the minimum possible energy E_0 has to be greater than 0.

A further rather interesting effect can be seen at the wave function in figure 2. Namely, the probability $|\psi(x)|^2$ is greater than 0 even if the total energy E is lower than the potential energy $V(x)$. This at the first glance very strange behavior which cannot be explained classically is basically caused by the fact that quantum mechanics postulates continuity of the wave function. In a similar way one observes the tunnel effect (c.f. section 3.2.4) at potential walls.

3.2 One-dimensional, time-dependent case

3.2.1 Schrödinger's equation

Schrödinger's equation in the one-dimensional, time-dependent case is

$$\left(-\frac{\hbar^2}{2m}\nabla^2 + V(x,t)\right)\Psi(x,t) = i\hbar\frac{\partial\Psi(x,t)}{\partial t}$$

with complex-valued wavefunction $\Psi(x,t)$.

3.2.2 Algorithm

To solve the one-dimensional, time-dependent SCHRÖDINGER equation the so-called *Goldberg-Schey-Schwartz*-algorithm is used.[15] The desired complex-valued wave function $\Psi(x,t)$ is formatted in a grid:

$$\begin{aligned} &x = j\epsilon,\ j = 0,1,\dots J, \\ &t = n\delta,\ n = 0,1,\dots, \end{aligned} \qquad \Psi(x,t) \cong \Psi_j^n. \tag{3.20}$$

Now we look for an iteration which calculates from the inital wave function Ψ_j^0 and the potential V_j^n the whole wave function Ψ_j^n:

$$\begin{aligned} &x = j\Delta t,\ j = 0,1,\dots J, \\ &t = n\Delta t,\ n = 0,1,\dots \end{aligned} \qquad \Psi(x,t) \cong \Psi_j^n$$

Iteration for Ψ_j^n:

1. $e_1 = 2 + 2\Delta t^2 V_1 - \frac{4i\,\Delta t^2}{\Delta t}$

   $e_j = 2 + 2\Delta t^2 V_j - \frac{4i\,\Delta t^2}{\Delta t} - \frac{1}{e_{j-1}}, \quad j = 2\dots J$

2. $\Omega_j^n = -\Psi_{j+1}^n + \left(\frac{4i\,\Delta t^2}{\Delta t} + 2\Delta t^2 V_j + 2\right)\Psi_j^n - \Psi_{j-1}^n, \quad j = 0\dots J$

3. $f_1^n = \Omega_1^n$ and $f_j^n = \Omega_j^n + \frac{f_{j-1}^n}{e_{j-1}}, \quad j = 2,\dots J$

4. $\Psi_{J-1}^{n+1} = -\frac{f_{J-1}^n}{e_{J-1}}$,

   $\Psi_j^{n+1} = \frac{\Psi_{j+1}^{n+1} - f_j^n}{e_j}, \quad j = J-2\dots 1$

5. $n = n+1$ and go to 1.

[15] For derivation see [5] p. 103ff.

For the wave function Ψ_j^n or $\Psi(x,t)$ being complex-valued it is reasonable to consider the visualization of complex numbers and complex functions. A very interesting discussing concerning this topic is written in chapter 1 of [7]. Additionally to the possibility of splitting the complex function into real and imaginary part the method of a color map is suggested. This color map associates every phase of the complex numbers with a certain color. At the same time the absolute value of the complex numbers is shown by the saturation of the color corresponding to the phase. This complex color map was used in the program in order to visualize the time-dependent complex-valued wave function properly.

In the program, the time-evolution of the wave function is put out as animation incrementing time t or alternatively as two-dimensional functions $\Psi(x,t)$ with time axis.

3.2.3 Potential box

At first the potential box is considered again, c.f. section 3.1.4. Let the inital wave function $\Psi(x,0)$ be a Gaussian wave packet

$$\Psi(x,0) = \exp\left[ikx - \frac{(x-x_0)^2}{2b_0^2}\right], \tag{3.21}$$

with k wave number, x_0 initial position and b_0 initial width. Incrementing time t yields a movement of the wave packet with velocity $v = \hbar k/m$ (fig. 4b and fig. 4e). The time-evolution also shows that the width of the wave packet gets bigger. Due to the boundary conditions $\Psi(0,t) = \Psi(a,t) = 0$ at the walls of the box the wave packet is reflected to the middle again at $x = 0$ and $x = a$ (fig. 4c and fig. 4e). These occuring reflexions at the walls lead to wild interferences at first. But after some time T the initial wave packet appears again. That is, there are destructive interferences after some time leading to periodicy of the time-evolution of the wave packet.[16]

3.2.4 Potential wall and tunnel effect

If a wave packet (3.21) hits a potential wall

$$V(x) = \begin{cases} V_0 & \text{für } |x| < a \\ 0 & \text{sonst,} \end{cases} \tag{3.22}$$

one can observe that the wave packet is split up into a reflected and a transmitted part. (fig. 5a and fig. 5b). A small subprogram is implemented in the program which allows to explicitly measure the reflexion and transmission coefficients R and D in respect of e.g. particle energy E, wall height V_0 or box width a. Fig. 5c shows the function $D(E,V_0)$. As one sees, the transmission coefficient D is directly proportional to the box height V_0. Rising the total energy E leads to a rising transmission coefficient D, nearly like the square root function.

[16] For a detailed discussion of this periodicy see [3] p. 146ff.

Interestingly to note the transmission coefficient D is still greater than 0 if the wave packet hits the wall with a total energy $E < V_0$! I.e. the probability that a particle with $E < V_0$ is measured after the wall is greater than 0 — the particle can *tunnel* the wall.

3.2.5 Harmonic oscillator

Figures 6a until 6c show that a wave packet (3.21) in a harmonic oscillator potential (c.f. section 3.1.5) oscillates from the left to the right again and again. That is the behavior of the particle here is very similar to the classical description.

3.3 Two-dimensional, time-independent case

Schrödinger equation :

$$\left(-\frac{\hbar^2}{2m}\nabla^2 + V(x,y)\right)\Psi(x,y) = E\Psi(x,y)$$

Solution: *centered space (CS) discretization and relaxation* yields

$$\psi(x,y) = \frac{\psi(x+h,y)+\psi(x-h,y)+\psi(x,y+h)+\psi(x,y-h)}{4-h^2F(x,y)}$$

with $F(x)$ from section 3.1.1.

3.4 Two-dimensional, time-dependent case

Schrödinger equation :

$$\left(-\frac{\hbar^2}{2m}\nabla^2 + V(x,y,t)\right)\Psi(x,y,t) = i\hbar\frac{\partial\Psi(x,y,t)}{\partial t}$$

$$\text{or} \quad \frac{\partial^2\Psi(x,y,t)}{\partial x^2}+\frac{\partial^2\Psi(x,y,t)}{\partial y^2}+i\frac{2m}{\hbar}\frac{\partial\Psi(x,y,t)}{\partial t}-\frac{2m}{\hbar}V(x,y,t)\Psi(x,y,t) = 0$$

Solution: *forward time, centered space (FTCS) algorithm.* Discretization and solving the obtained system yields recursion for $\Psi(x,y,t+\Delta t) = \Psi_r(x,y,t+\Delta t) + i\Psi_i(x,y,t+\Delta t)$

$$\Psi_r(x,y,t+\Delta t) = \frac{c_2V(x,y,t)\Psi_r^{\partial^2}(x,y,t)+c_1\Psi_i^{\partial^2}(x,y,t)+c_1^2\Psi_r(x,y,t)-c_1c_2V(x,y,t)\Psi_i(x,y,t)}{c_1^2+c_2^2V(x,y,t)^2},$$

$$\Psi_i(x,y,t+\Delta t) = \frac{c_2V(x,y,t)\Psi_i^{\partial^2}(x,y,t)-c_1\Psi_r^{\partial^2}(x,y,t)+c_1^2\Psi_i(x,y,t)-c_1c_2V(x,y,t)\Psi_r(x,y,t)}{c_1^2+c_2^2V(x,y,t)},$$

$$\Psi_r^{\partial^2}(x,y,t) = \frac{\Psi_r(x+\Delta x,y,t)+\Psi_r(x-\Delta x,y,t)+\Psi_r(x,y+\Delta y,t)+\Psi_r(x,y-\Delta y,t)-4\Psi_r(x,y,t)}{(\Delta x)^2},$$

$$\Psi_i^{\partial^2}(x,y,t) = \frac{\Psi_i(x+\Delta x,y,t)+\Psi_i(x-\Delta x,y,t)+\Psi_i(x,y+\Delta y,t)+\Psi_i(x,y-\Delta y,t)-4\Psi_i(x,y,t)}{(\Delta x)^2}$$

with $c_1 = -\frac{2m}{\hbar\Delta t}$ and $c_2 = \frac{2m}{\hbar^2}$.

fig. 1: *Potentialkasten* (section 3.1.4). Oben Potential mit Energie-Eigenwerten, unten Wellenfunktion.
fig. 2: *Harmonischer Oszillator* (section 3.1.5). Die Energie-Eigenwerte sind äquidistant, vgl. equation (3.19).

fig. 3: *Farbcode* für komplexe Zahlen (section 3.2.2).
fig. 4a: *Zeitlicher Verlauf* eines Wellenpakets im *Potentialkasten* (section 3.2.3).
fig. 4b: Das Wellenpaket zerfließt.
fig. 4c: Reflexion an den Wänden.
fig. 4d: Anfangszustand wiederhergestellt.
fig. 4e: $\Psi(x,t)$ als 3D-Plot. Die Ortsachse verläuft nach rechts, die Zeitachse nach hinten.
fig. 5a: Wellenpaket im *Barrierepotential*. Aufspaltung in Reflexion und Transmission.
fig. 5b: 3D-Plot von $\Psi(x,t)$. Die Ortsachse verläuft nach rechts, die Zeitachse nach hinten.
fig. 5c: Funktion $D(E,h)$.
fig. 6a: *Harmonischer Oszillator*. Wellenpaket auf ist auf der linken Seite.
fig. 6b: Wellenpaket befindet sich rechts. Nun wird das Wellenpaket wieder nach links laufen, hierauf wieder nach rechts, usw.. . .
fig. 6c: 3D-Plot von $\Psi(x,t)$. Die Ortsachse verläuft nach rechts, die Zeitachse nach hinten bzw. oben.

3.5 Radial, time-independent case

The solution of the radial, time-independent SCHRÖDINGER equation is fairly difficult. Analytical solutions only exist for potentials with many symmetry properties. Whatsoever, the SCHRÖDINGER equation

$$\underbrace{\left(\frac{\mathbf{p}^2}{2m} + V(r)\right)}_{H} \psi(\mathbf{r}) = E\psi(\mathbf{r}) \tag{3.23}$$

for radial potentials $V(\mathbf{r}) = V(r)$ can be solved numerically quite good.[17] The wave function $\psi(\mathbf{r})$ is split into the spherical eigenfunctions $Y_{l,m}(\Theta, \varphi)$ (given by recursion formula) and the radial wave function $\psi_{rad_{n,l}}(r)$

$$\psi(\mathbf{r}) = Y_{l,m}(\Theta, \varphi)\frac{\psi_{rad_{n,l}}(r)}{r}. \tag{3.24}$$

Scaling $\frac{\hbar^2}{m} = 1$ makes it possible to reduce (3.23) to a one-dimensional problem which can be solved by NUMEROV's algorithm (c.f. section 3.1.1) again:

$$\frac{\mathrm{d}^2}{\mathrm{d}r^2}\psi_{rad}(r) + \underbrace{\left(2(E - V(r)) - \frac{l(l+1)}{r^2}\right)}_{F(r)} \psi_{rad}(r) = 0. \tag{3.25}$$

4 Implementation

Still to be translated into English...

In diesem Abschnitt soll nun auf die konkrete Implementierung der in section 3 besprochenen Verfahren eingegangen werden.

4.1 Choice of programming language (Java)

Die Programme wurden in der Programmiersprache *Java*, die mit C++ die zur Zeit weltweit am weitesten verbreitete Programmiersprache ist, realisiert. Es ist der Trend zu beobachten, dass immer mehr Menschen Java lernen und dass Java C++ als am häufigsten verwendete Programmiersprache womöglich bald ablösen wird[18]. Die wichtigsten Vorteile von Java sind:

- Die Sprache ist äußerst objektorientiert aufgebaut. Der Compiler 'zwingt' den Entwickler nahezu zu objektorientierter Programmierung. Objektorientierte Programmierung hat sich in den letzten Jahren als äußerst erfolgreich erwiesen.
- Die Programme sind plattformunabhängig.

[17] For a more detailed description see [5] p. 50ff.
[18] Manche behaupten, dies sei schon heute der Fall.

- Es kann auf eine sehr umfangreiche Klassenbibliothek zurückgegriffen werden, die auf jedem System, auf dem Java installiert ist, verfügbar ist.
- Es lassen sich ohne allzu großen Aufwand anspruchsvolle interaktive grafische Benutzeroberflächen erstellen.

Konkret wurden sogenannte *Java Applets* programmiert, deren wichtigster Vorteil darin besteht, dass sie direkt in Webseiten integriert werden können, wodurch eine Installation entfällt.

4.2 Used packages

Um die Programmierung der Applets zu verkürzen, wurde auf einige Pakete zurückgegriffen. So werden die Ausgaben dreidimensionaler Funktionen, welche ja in den Abschnitten 3.2 und 3.5 benötigt werden, von den Paketen `Java3D`[19] und `VisAD`uebernommmen. Diese inzwischen recht weit entwickelten Pakete zeichnen sich v.a. durch Plattformunabhängigkeit und durch sehr hohe Flexibilität aus. So können die dreidimensionalen Ausgaben in Echtzeit vom Benutzer berechnet werden. Sehr nützlich ist auch, dass Zoomen, Drehen und Verschieben (jeweils direkt mit der Maus) automatisch unterstützt wird. Die Abbildungen 4e, 5b und 6c stellen einige von `Java3D` und `VisAD` erzeugten Beispiele dar.

Zusätzlich wird ein Paket zum Rechnen mit komplexen Zahlen für die in den Abschnitten 3.2.2 und 3.5[20] beschriebenen Algorithmen benötigt. Hierzu wurde auf das Paket `JavaSci`[21] zurückgegriffen, weil es eine sehr umfangreiche Auswahl an weiteren mathematischen Hilfsmitteln enthält, die evtl. für eine Weiterentwicklung des Projekts notwendig sein könnten.

4.3 Parameter input and user-friendliness

Bei der Implementierung der Programme wurde auf sehr hohe Flexibilität Wert gelegt, d.h. nahezu alle für die Berechnungen benötigten Parameter sind vom Anwender in Echtzeit regelbar. Damit das Programm dennoch nicht unübersichtlich wird, wurde es in eine sehr anwenderfreundliche grafische Benutzeroberfläche von Paul Falstad mit dessen freundlicher Genehmigung eingebettet. Die Parameter lassen sich somit mit Schiebereglern, Eingabefeldern oder Drop-Down-Listen steuern. Zusätzlich können viele Parameter direkt in der Anzeigefläche des Programms mit der Maus – z.B. durch Ziehen oder Klicken – bestimmt werden.

Für das Programm ist v.a. auch ein didaktischer Nutzen angedacht. Hilfetexte sollen nicht wie meist üblich ausschließlich in separaten Fenstern oder Dateien abrufbar sein, sondern sie sollen vielmehr direkt während des Ausführens des Programms angezeigt werden. Hierbei wird nicht eine große Menge Text auf ein Mal angezeigt, sondern es werden Textabschnitte aus nur

[19] Sun Microsystems, Inc.: *Java3D.* `http://java.sun.com/products/java-media/3D/`.

[20] Die Komplexität wird hier von den Kugelflächenfunktionen $Y_{l,m}(\Theta, \varphi)$, siehe [5] S. 41ff., verursacht.

[21] Hale, Mark et al.: *JavaSci – A science API for Java.* `http://jsci.sourceforge.net/`.

wenigen Zeilen durchlaufen. Diese kleinen Textstücke sollen den Anwender Schritt für Schritt durch das Programm führen und die physikalischen Hintergründe an den Kenntnisstand des Anwenders angepasst erläutern. Sehr viel Wert soll dabei auf eine hohe Interaktivität gelegt werden, beispielsweise durch eine große Auswahl an verschiedenen Touren, die gewählt werden können, oder durch ausführliche Anleitungen, mit welchen der Benutzer verschiedene Situationen und Effekte *selbst* 'zusammenbauen' kann. Um so beispielsweise zu zeigen, dass ein Teil eines Wellenpakets mit $E < V_0$ (vgl. section 3.2.4) die Potentialbarriere überwinden kann, wird der Anwender, z.B. ein Schüler, nicht einfach vor einen statischen Ausdruck des Transmissionskoeffizienten gesetzt, sondern ihm wird eine step-by-step Anleitung angeboten, nach der er die Potentialbarriere selbst zeichnen kann, das Wellenpaket selbst bestimmen und auf die Barriere losschicken kann und die Bewegung *seines* Wellenpakets auf *seine* Potentialbarriere selbst beobachten kann.

5 Status of development and download availability

Status of development

At the moment, i.e. in May 2004, all of the above cases are implemented and the program does fairly well.

Download

All programs can be downloaded for free from `http://javapsi.sourceforge.net/`. This page also contains some additional information on the project.

6 Future aims

Still to be translated into English...

Zur Zeit konzentriert sich die Arbeit darauf, die in section 3 beschriebenen Verfahren vollständig in Java Applets zu verwirklichen. Später wäre es beispielsweise möglich, verschiedene Ansätze zur physikalischen Funktionsweise eines Quantencomputers[22] mit Java Applets zu visualisieren. Zwar existieren schon Programme, die Quantencomputer *simulieren*, aber Programme, die Quantencomputer *visualisieren*, finden sich kaum. Während Simulationen von Quantencomputern dem Testen von Algorithmen für Quantencomputer dienen, ließe sich mit Visualisierungen von Quantencomputern unter Umständen die physikalische Funktionsweise optimieren. Genau hierin liegt die Herausforderung, denn Algorithmen für Quantencomputer existieren schon seit geraumer Zeit — die praktische Verwirklichung eines Quantencomputers steckt jedoch noch in den Kinderschuhen.

[22] Sehr interessant zu diesem Thema ist Kapitel 7 (S. 277ff) in Nielsen, Michael A.: *Quantum computation and quantum information.* Cambridge, 2001.

Eine weiterer Gedanke ist, das in section 2.4 vorgestellte DIRAC-Modell zu visualisieren. Eine derartige Veranschaulichung am Computer existiert meines Wissens noch nicht. Das größte Problem bei der Darstellung des DIRAC-Modells stellt wahrscheinlich der HILBERT-Raum dar, den man sich ja nicht vorstellen kann. Trotzdem wäre es sicher sehr interessant, Möglichkeiten zu finden, mit denen das DIRAC-Modell visualisiert werden könnte.

Langfristig ist auch eine Behandlung von Mehrteilchensystemen denkbar. Mit statistischen Hilfsmitteln lassen sich Wechselwirkungen zwischen mehreren Teilchen nämlich recht gut am Computer simulieren.

7 Acknowledges

Special thanks to Paul Falstad, Andreas Greiner, Christian Hoffmann, Martin Kamp, Wolfgang Kinzel, Siegfried Krewald, Thomas Michelitsch und Nathan Urban for interesting explanations and discussions. Further I'd like to thank the Rotary-Club Nürnberg-Erlangen for the computer sponsoring. Last but not least I thank Christoph Groth for recommending Linux, Emacs and LaTeX.

References

[1] DIRAC, PAUL A. M.: *The Principles of Quantum Mechanics*. Oxford, 1988.

[2] GASIOROWICZ, STEPHEN: *Quantenphysik*. München, 1989.

[3] KINZEL, WOLFGANG ET AL.: *Physics by Computer*. Heidelberg, 1998.

[4] PRESS, WILLIAM H. ET AL.: *Numerical Recipes in C*. Cambridge, 2002.

[5] SCHNAKENBERG, JÜRGEN: *Algorithmen in der Quantentheorie und Statistischen Physik*. Ulmen, 1995.

[6] SCHWABL, FRANZ: *Quantenmechanik*. Berlin, 2002.

[7] THALLER, BERND: *Visual Quantum Mechanics*. New York, 2000.