Wolfgang Kinzel Georg Reents

Physics by Computer

Programming Physical Problems Using Mathematica® and C

Translated by Martin Clajus and Beverly Freeland-Clajus

With 88 Figures, 33 Exercises and a CD-ROM
Containing Programs and Graphics Routines for PCs
and Workstations; Mathematica® 3.0 compatible

Springer

Springer
Berlin
Heidelberg
New York
Barcelona
Budapest
Hong Kong
London
Milan
Paris
Santa Clara
Singapore

4.4 Solitons

So far, we have only considered differential equations for functions that depend on *one* variable (time or position). Now we want to study a wave that moves and possibly changes its shape as a function of time. In this case, the time dependence is coupled to the spatial dependence and one gets a partial differential equation.

In this section, we want to deal with the *Korteweg–de Vries equation* (KdV equation) and make an exception by numerically reproducing a solution that can be obtained analytically as well. The KdV equation is a nonlinear differential equation with special solutions, the so-called solitons, that exhibit a number of surprising properties. For example, two solitons can move towards one another, penetrate one another and then move on at their original velocity without changing shape.

Physics

In the theory of water waves in shallow channels, the deviation $u(x,t)$ from the mean water level at time t and position x is described by a nonlinear partial differential equation. In dimensionless and suitably normalized form, the KdV equation can be written as

$$\frac{\partial u}{\partial t} = 6u\frac{\partial u}{\partial x} - \frac{\partial^3 u}{\partial x^3} \; . \tag{4.37}$$

The contribution from the term $u\partial u/\partial x$ increases fourfold if the deviation is doubled, which means that the equation is nonlinear.

Among the many solutions of the KdV equation, there is one that is rather characteristic, namely

$$u\,(x,t) = -2\,\mathrm{sech}^2\,(x-4t) \; , \tag{4.38}$$

with $\mathrm{sech}\,x = 1/\cosh x$. This function $u(x,t)$ describes a wave trough that moves to the right at a velocity $v = 4$, not changing its shape in the process. Such a solution is called a soliton.

More complex solutions are known as well, though. The mathematical theory for this is not simple, which is why we only state the result here. The initial state,

$$u\,(x,0) = -N\,(N+1)\;\mathrm{sech}^2\,(x) \; , \tag{4.39}$$

results in N solitons that propagate at different velocities. For example, for $N = 2$, one finds

$$u\,(x,t) = -12\frac{3+4\cosh\,(2x-8t)+\cosh\,(4x-64t)}{[3\cosh\,(x-28t)+\cosh\,(3x-36t)]^2} \; . \tag{4.40}$$

Figure 4.8 shows that this solution contains two solitons. A wave with a large amplitude has overtaken a broader wave with a small amplitude. At time

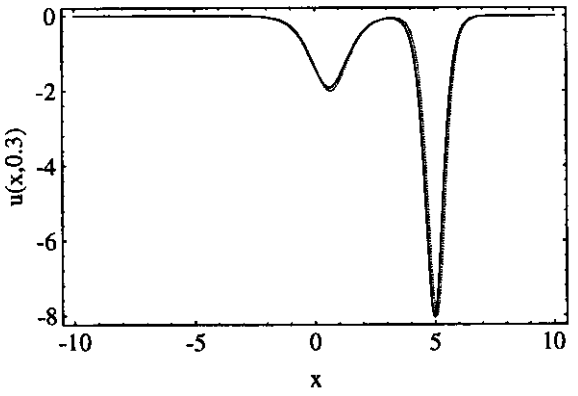

Fig. 4.8. Two solitons move away from one another. The numerical solution (*solid curve*) and the exact one (*dotted curve*) are in good agreement ($t = 0.3$, $\mathrm{d}x = 0.18$, $\mathrm{d}t = 0.002$)

$t = 0$, the two overlap to form the wave packet (4.39), whereas after their encounter both solitons have resumed their original shapes.

We would also like to mention an interesting link to quantum mechanics: The Schrödinger equation with the potential $V(x) = -N(N+1)\;\mathrm{sech}^2(x)$ has N bound states, from which one can construct the solutions of the KdV equation by using the methods of inverse scattering theory. Since the explanation is too complicated for the scope of this book, we refer the reader to the literature about solitons.

Algorithm and Results

In order to numerically solve the KdV equation (4.37), we must first discretize the space and time coordinates x and t,

$$u_n^j = u\,(j\mathrm{d}x, n\mathrm{d}t) \; . \tag{4.41}$$

Here j and n are integers, and $\mathrm{d}x$ and $\mathrm{d}t$ are the step sizes in the x and t coordinates respectively. We want to demonstrate what happens if we write (4.37) too naively as a difference equation. For example, we could consistently use the so-called forward-two-point formula for each derivative $f'(x_k)$, i.e., approximate $f'(x_k)$ by $(f(x_k+h)-f(x_k))/h$. For the partial derivatives this yields

$$\frac{\partial u}{\partial t} \;\to\; \frac{1}{\mathrm{d}t}\left(u_{n+1}^j - u_n^j\right) \; , \quad \frac{\partial u}{\partial x} \;\to\; \frac{1}{\mathrm{d}x}\left(u_n^{j+1} - u_n^j\right) \; . \tag{4.42}$$

By applying this rule three times we obtain

$$\frac{\partial^3 u}{\partial x^3} \;\to\; \frac{1}{(\mathrm{d}x)^3}\left(u_n^{j+3} - 3u_n^{j+2} + 3u_n^{j+1} - u_n^j\right) \; . \tag{4.43}$$

With these substitutions, we solve (4.37) for u_{n+1}^j with the result

$$u_{n+1}^j = u_n^j + \mathrm{d}t\left(6u_n^j\frac{u_n^{j+1}-u_n^j}{\mathrm{d}x} - \frac{u_n^{j+3}-3u_n^{j+2}+3u_n^{j+1}-u_n^j}{(\mathrm{d}x)^3}\right) \; . \tag{4.44}$$

This equation is easily programmable. In *Mathematica*, we represent $\{u_n^j\}_{j=0}^{\max}$ by a list with max + 1 elements. To initialize it, we use (4.39) with $N = 2$:

```
ustart:=Table[-6 Sech[(j-max/2)dx]^2//N,{j,0,max}]
```

and assume periodic boundary conditions. We obtain the shifted lists $\{u_n^{j+k}\}$ for $k = 1$, 2, and 3 via `uplus[k]=RotateLeft[u,k]` and can then formulate the integration step $\{u_n^j\} \to \{u_{n+1}^j\}$ as follows:

```
step[u_]:=(Do[uplus[k]=RotateLeft[u,k],{k,3}];
       u+dt(6 u(uplus[1]-u)/dx -
         (uplus[3]-3 uplus[2]+3 uplus[1]-u)/dx^3))
```

`plot2[i_:3]` lets us plot the result after `i` integration steps, using $dx = 0.05$ and $dt = 0.02$.

```
plot2[i_:3]:= (dx=0.05; dt=0.02; upast=ustart; time=0;
 Do[upres=step[upast];
    upast=upres;Print["Time ",time=time+dt],{i}] ;
    xulist = Table[{(j-max/2)dx,upres[[j]]},{j,0,max}];
    ListPlot[xulist, PlotJoined->True, PlotRange->All])
```

Figure 4.9 shows the result. After just three time steps, $u(x,t)$ exhibits oscillations, which are caused by numerical inaccuracies and "explode" after just two more steps. This, of course, is unphysical and can be blamed on the bad algorithm.

What went wrong? We have committed several errors, which we can avoid by the following modifications:

1. By a better discretization of the derivatives, the error in the approximation of $\partial/\partial t$ and $\partial^n/\partial x^n$ can be reduced.
2. The use of averages stabilizes the algorithm.
3. The step size dx chosen must always be large in comparison to dt.

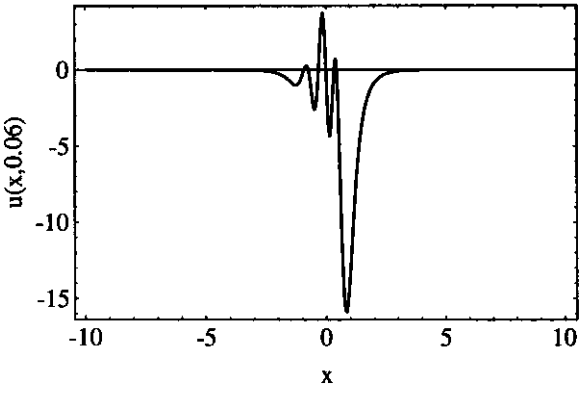

Fig. 4.9. Integration using `step[u]` and step sizes $dx = 0.05$, $dt = 0.02$. After just three integration steps, one obtains an unphysical result. The reader should compare this to Fig. 4.8

Point 1 can be illustrated by the following argument: Consider the Taylor expansion of $u(x,0)$ about $x = 0$, using the notation $u_0' = (\partial/\partial x)u(x,0)|_{x=0}$ etc.,

$$u_{\pm 1} = u_0 \pm dx u_0' + \frac{(dx)^2}{2} u_0'' \pm \frac{(dx)^3}{6} u_0''' + \cdots .$$

Then,

$$u_0' = \frac{u_1 - u_0}{dx} - \frac{dx}{2} u_0'' + \cdots ,$$
$$u_0' = \frac{u_1 - u_{-1}}{2dx} - \frac{(dx)^2}{6} u_0''' + \cdots . \tag{4.45}$$

One sees that in the second case, u_0' can be calculated in such a way that $(dx/2)u_0''$ cancels exactly. This reduces the error by one order in dx. In a similar way, one can reduce the error of the higher derivatives by a suitable choice of the coefficients. One obtains

$$u_0'' = \frac{u_1 - 2u_0 + u_{-1}}{dx^2} + \mathcal{O}(dx^2) ,$$
$$u_0''' = \frac{u_2 - 2u_1 + 2u_{-1} - u_{-2}}{2dx^3} + \mathcal{O}(dx^2) . \tag{4.46}$$

Thus, with respect to the quality of the approximation, the algorithm used in (4.44) gave away one order in dx and dt for all derivatives.

To illustrate points 2 and 3, we want to use a simple differential equation. In particular, we choose

$$\frac{\partial u}{\partial t} = -v \frac{\partial u}{\partial x}$$

with a constant velocity v. We discretize this equation in the form

$$\frac{u_{n+1}^j - u_n^j}{dt} = -v \frac{u_n^{j+1} - u_n^{j-1}}{2dx} , \tag{4.47}$$

where we have already substituted the improved version (4.45) for $\partial/\partial x$. Solving for the components $(u_{n+1}^1, u_{n+1}^2, \ldots)$, which we combine into $\boldsymbol{u}_{n+1}$, we can write (4.47) as a matrix equation

$$\boldsymbol{u}_{n+1}^{\mathsf{T}} = \mathsf{M}\boldsymbol{u}_n^{\mathsf{T}} \tag{4.48}$$

with a tridiagonal matrix M. Obviously, the solution of (4.48) is $\boldsymbol{u}_n^{\mathsf{T}} = \mathsf{M}^n \boldsymbol{u}_0^{\mathsf{T}}$, from which we can see that the magnitude of the eigenvalues of M is critical for the stability of the algorithm. For the eigenmodes $\boldsymbol{w}$ of matrices of this type we use a Fourier ansatz of the form $w^j = \exp(ikjdx)$ with a wave vector k whose possible values are determined by the boundary conditions. If we choose an eigenmode $\boldsymbol{w} = \boldsymbol{w}_0$ as our initial state, and designate the corresponding eigenvalue as a, then applying the matrix M n times yields

$$w_n^j = a^n \exp(ikjdx) . \tag{4.49}$$

If we insert this into (4.47), we obtain an equation from which we can determine the amplitude a:

$$\frac{a^{n+1}-a^n}{\mathrm{d}t} = -va^n \frac{\mathrm{e}^{\mathrm{i}k\mathrm{d}x}-\mathrm{e}^{-\mathrm{i}k\mathrm{d}x}}{2\mathrm{d}x} . \tag{4.50}$$

The solution is

$$a = 1 - \mathrm{i}\frac{v\mathrm{d}t}{\mathrm{d}x}\sin(k\mathrm{d}x) . \tag{4.51}$$

Thus, a is complex, with a magnitude which is always greater than one, except for $k = 0$. This means that any initial mode which does not happen to be constant grows exponentially with time. The algorithm is unusable!

A seemingly insignificant change can stabilize the algorithm, however. In (4.47) we replace u_n^j by the average,

$$u_n^j \to \frac{1}{2}\left(u_n^{j+1}+u_n^{j-1}\right) . \tag{4.52}$$

This gives us

$$u_{n+1}^j = \frac{1}{2}\left(u_n^{j+1}+u_n^{j-1}\right) - \frac{v\mathrm{d}t}{2\mathrm{d}x}\left(u_n^{j+1}-u_n^{j-1}\right) , \tag{4.53}$$

and the amplitude equation yields

$$a = \cos(k\mathrm{d}x) - \mathrm{i}\frac{v\mathrm{d}t}{\mathrm{d}x}\sin(k\mathrm{d}x) . \tag{4.54}$$

Now we have $|a| \le 1$ for

$$|v|\,\mathrm{d}t \le \mathrm{d}x . \tag{4.55}$$

This inequality is called the Courant condition. It states that perturbations only remain stable if the time step $\mathrm{d}t$ is chosen so that it is smaller than the propagation time $\mathrm{d}x/|v|$. Or, from the opposite perspective, the spatial discretization chosen must not be too fine.

According to (4.45) it is better to discretize the time derivative symmetrically as well:

$$u_{n+1}^j - u_{n-1}^j = -\frac{v\mathrm{d}t}{\mathrm{d}x}\left(u_n^{j+1}-u_n^{j-1}\right) . \tag{4.56}$$

For the amplitude, this yields the equation

$$\left(a - \frac{1}{a}\right) = -2\mathrm{i}\frac{v\mathrm{d}t}{\mathrm{d}x}\sin k\mathrm{d}x . \tag{4.57}$$

The solution,

$$a = -\mathrm{i}\frac{v\mathrm{d}t}{\mathrm{d}x}\sin k\mathrm{d}x \pm \sqrt{1-\left(\frac{v\mathrm{d}t}{\mathrm{d}x}\sin k\mathrm{d}x\right)^2} , \tag{4.58}$$

shows that $|a| = 1$ if the Courant condition (4.55) holds. Here, too, $\mathrm{d}t$ has to be chosen to be sufficiently small compared to $\mathrm{d}x$.

Equation (4.56) now contains three time steps, $n+1$, n, and $n-1$. To calculate the future, we have to know present and past. In the first step, we only know the present, so the future has to be calculated in one step, e.g., using (4.53). All other steps can then be done according to (4.56).

Now we want to use these insights for the numerical solution of the KdV equation. We substitute

$$\frac{\partial u}{\partial t} \to \frac{u_{n+1}^j - u_{n-1}^j}{2\mathrm{d}t} ,$$

$$\frac{\partial u}{\partial x} \to \frac{u_n^{j+1} - u_n^{j-1}}{2\mathrm{d}x} ,$$

$$u \to \frac{1}{3}\left(u_n^{j+1}+u_n^j+u_n^{j-1}\right) ,$$

$$\frac{\partial^3 u}{\partial x^3} \to \frac{1}{2(\mathrm{d}x)^3}\left(u_n^{j+2}-2u_n^{j+1}+2u_n^{j-1}-u_n^{j-2}\right) .$$

We call $n-1$ past, n present, and $n+1$ future. Then a time step is given by

```
step2[u_,w_]:=(up1=RotateLeft[u]; up2=RotateLeft[up1];
               um1=RotateRight[u];um2=RotateRight[um1];
               w+dt(2(um1+u+up1)*(up1-um1)/dx -
                    (up2-2 up1+2 um1-um2)/dx^3 ) )
```

With `step2`, we generate `ufut`, the list for the future, if we insert the present (list `upres`) and the past (list `upast`) as arguments. By using `plot3[i_:9]`, this is iterated and plotted after `i` time steps, a smooth curve through the x–u values having been drawn first via `Interpolation[xulist]`.

```
plot3[i_:9]:=
   (Do[ufut=step2[upres,upast];
       upast=upres;upres=ufut; time = time+dt, {i}];
    Print["time ",time];
    xulist=Table[{(j-max/2)*dx,ufut[[j+1]]},{j,0,max}];
    uu=Interpolation[xulist];
    Plot[uu[x],{x,-10.,10.},PlotRange->All,
               Frame -> True, Axes -> None ] )
```

Figure 4.8 shows the result for the initial state $u(x,0) = -6\,\mathrm{sech}^2(x)$. There is good agreement between the numerical solution and the exact one, which is shown by the dotted curve. The wave trough $u(x,0)$ turns into two solitons that move to the right at different velocities. This becomes particularly evident from the three-dimensional plot (Fig. 4.10) of $u(x,t)$ and the corresponding contour plot (Fig. 4.11). For negative times, the fast soliton with the large amplitude approaches the slow one. At $t = 0$, the two combine to form a single wave trough, and after the encounter both resume their original shape. While being overtaken, the small soliton experiences a delay, whereas the large one is accelerated. This can be seen particularly clearly in the contour plot of Fig. 4.11.

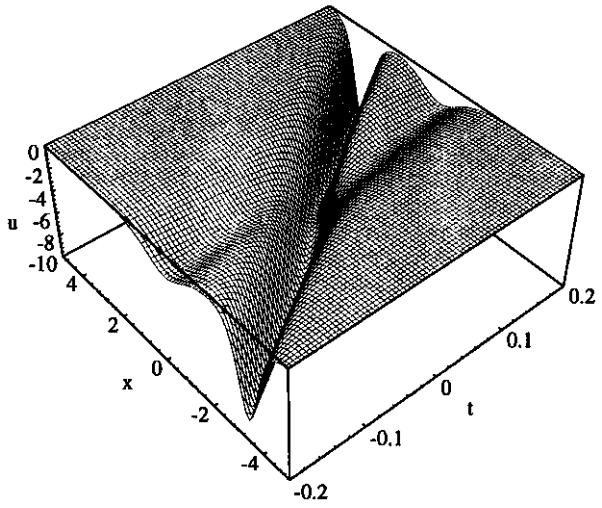

Fig. 4.10. The two solitons from Fig. 4.8, shown in a three-dimensional plot for different times. A large, fast soliton passes a small, broad one

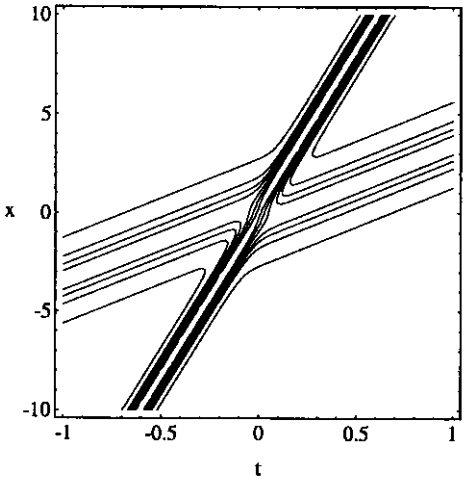

Fig. 4.11. The same data as in Fig. 4.10, but as a contour plot and with different ranges in space and time. While being passed, the small, broad soliton is delayed, whereas the large one is accelerated in the process

The analytic solutions mentioned at the beginning of this section evolve from an initial state $-N(N+1)\,\mathrm{sech}^2 x$, which decays to N solitons. Thus, the amplitudes -2 and -6 lead to the one- and two-soliton solution, respectively. But what happens to initial states whose amplitude falls in between? Figure 4.12 shows the numerical solution for the initial state $u(x,0) = -4\,\mathrm{sech}^2 x$. In addition to the soliton that moves to the right, additional waves disperse to the left.

It is self-evident that the quality of the numerical approximation can be improved by reducing the step size $\mathrm{d}x$. The effort, however, increases signif-

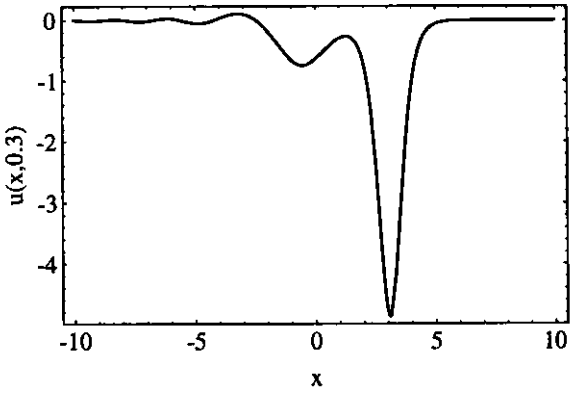

Fig. 4.12. Results similar to those in Fig 4.8, but for an initial state $u(x,0) = -4\,\mathrm{sech}^2 x$. In addition to the soliton moving to the right, waves are radiated

icantly in the process. Obviously, the length of the lists $\{u_n^j\}_{j=0}^{\max}$ increases as $1/\mathrm{d}x$. The following exercise is intended to demonstrate that, at the same time, the maximum $\mathrm{d}t$ has to be reduced in proportion to $(\mathrm{d}x)^3$, for the sake of stability.

Exercises

The stability of the algorithm programmed in `step2[u_,w_]` is limited significantly by the discretized third derivative.

1. Remove the nonlinear term from the full KdV equation, i.e., investigate the equation

$$\frac{\partial u}{\partial t} = -\frac{\partial^3 u}{\partial x^3}\,. \tag{4.59}$$

The discretization used here yields

$$u_{n+1}^j = u_{n-1}^j - \frac{\mathrm{d}t\left(u_n^{j+2} - 2u_n^{j+1} + 2u_n^{j-1} - u_n^{j-2}\right)}{(\mathrm{d}x)^3}\,. \tag{4.60}$$

Investigate the stability of this algorithm by using the ansatz (4.49) in (4.60). Prove that, analogously to (4.57), this leads to the amplitude equation

$$a = \frac{1}{a} - \frac{2\mathrm{i}\mathrm{d}t\left[\sin(2k\mathrm{d}x) - 2\sin(k\mathrm{d}x)\right]}{(\mathrm{d}x)^3}\,.$$

Numerically determine the maximum μ of $|\sin(2k\mathrm{d}x) - 2\sin(k\mathrm{d}x)|$ ($\mu \simeq 2.6$), and show that, in combination with the requirement $|a| \le 1$, this yields the stability condition

$$\mathrm{d}t \le \frac{1}{\mu}(\mathrm{d}x)^3\,. \tag{4.61}$$

2. Use the algorithm `step2[u,w]` for the integration, and choose the initial condition $u(x,0) = -6\,\mathrm{sech}^2 x$. Vary $\mathrm{d}x$ between 0.1 and 0.2 and determine the stability limit with respect to $\mathrm{d}t$, for a given $\mathrm{d}x$, i.e., approximately determine the maximum $\mathrm{d}t$ allowed as a function of $\mathrm{d}x$. Confirm (4.61) by displaying the result in a log–log plot.

Literature

Baumann G. (1996) Mathematica in Theoretical Physics: Selected Examples from Classical Mechanics to Fractals. TELOS, Santa Clara, CA

Crandall R.E. (1991) Mathematica for the Sciences. Addison-Wesley, Redwood City, CA

4.5 Time-dependent Schrödinger Equation

If a particle moves in a box without friction or any other force, its motion is changed only by reflection off the walls. In one dimension, therefore, it moves back and forth regularly. This classical picture, which is based on the idea of a pointlike mass with precisely defined position and momentum, no longer holds true if the box is microscopically small. Instead, we have to describe the particle by a wave function. In the quantum mechanics course one learns that the stationary states in this case are standing waves. But what happens to an initially localized wave packet that moves towards the walls of the box?

This problem has both analytic and numerical aspects. We will use the expansion in terms of eigenfunctions in order to make statements about symmetries, recurrence times, and other characteristic length and time scales. On the other hand, we want to use the time-dependent Schrödinger equation to demonstrate how to solve partial differential equations numerically by using an implicit method. This involves inverting a tridiagonal matrix, for which there is a fast numerical method. Additionally, care has to be taken when discretizing the time evolution of the quantum-mechanical state that the normalization of the wave function does not change with time.

We will see that the seemingly simple and well-understood textbook example exhibits a surprisingly complex behavior. The wave packet disperses, and wild interference patterns arise from which smooth wave packets suddenly reemerge. Finally, the entire process repeats periodically with time. The cover of this book shows what complexity and beauty can arise from elementary quantum mechanics.

Physics

Let $\psi(x,t)$ be the complex-valued wave function of a particle with mass m which moves in one dimension in a potential $\tilde{V}(x)$. Then $|\psi(x,t)|^2\mathrm{d}x$ is the probability of finding the particle in the interval $[x, x+\mathrm{d}x]$ at time t. The time dependence of $\psi(x,t)$ is described by the Schrödinger equation in its coordinate representation:

$$\mathrm{i}\hbar\frac{\partial\psi}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\psi}{\partial x^2} + \tilde{V}(x)\,\psi\,. \tag{4.62}$$

In order to put this equation into a dimensionless form, we normalize the time by t_0 and the position by x_0:

$$\mathrm{i}\frac{\hbar}{t_0}\frac{\partial\psi}{\partial(t/t_0)} = -\frac{\hbar^2}{2mx_0^2}\frac{\partial^2\psi}{\partial(x/x_0)^2} + \tilde{V}(x)\,\psi\,. \tag{4.63}$$

Now we choose t_0 and x_0 such that the following equation holds:

$$\hbar t_0 = 2mx_0^2\,. \tag{4.64}$$

Now if we set $\tilde{V}(x)t_0/\hbar = V(x/x_0)$ and express position and time in units of x_0 and t_0, we obtain the dimensionless equation

$$\mathrm{i}\frac{\partial\psi}{\partial t} = H\psi = \left[-\frac{\partial^2}{\partial x^2} + V(x)\right]\psi\,. \tag{4.65}$$

H is the normalized Hamiltonian of the particle. There are two ways to numerically solve this equation. First, we can calculate the eigenstates and eigenvalues of the stationary equation $H\psi = E\psi$, expand $\psi(x,0)$ in terms of these states, and then specify a series representation for $\psi(x,t)$. Second, we can directly integrate the time-dependent equation as shown in the previous section. The second method can even be used for problems for which the first one fails, which is why we want to describe that method in the algorithm section. If, on the other hand, the eigenstates are known, one can use the analytic ansatz to directly derive at least some properties of the wave function $\psi(x,t)$.

We want to study a particle in a box with infinitely high walls. As symmetry considerations will play a significant role in what follows, we position the coordinate system in such a way that this symmetry can be expressed easily. With these considerations, the potential $V(x)$ takes the form

$$V(x) = \begin{cases} 0 & \text{for} \quad -\frac{1}{2} \le x \le \frac{1}{2}\,, \\ \infty & \text{otherwise}\,. \end{cases} \tag{4.66}$$

Here, the coordinate x is expressed in units of the box's width a, and the energy E in units of $\hbar^2/2ma^2$. The energies E_n of the stationary states $\psi_n(x)$ are known from the quantum mechanics course:

$$\begin{aligned} E_n &= n^2\pi^2\,, \quad \text{where} \quad n = 1,2,3,\ldots\,, \\ \psi_n(x) &= \begin{cases} \sqrt{2}\cos(n\pi x) & \text{for } n \text{ odd} \\ \sqrt{2}\sin(n\pi x) & \text{for } n \text{ even} \end{cases}, \quad -\tfrac{1}{2} \le x \le \tfrac{1}{2}\,. \end{aligned} \tag{4.67}$$

These states are standing waves whose magnitude does not change as a function of time. The particle's mean position in these states, $\langle x\rangle = 0$, does not change with time either.