Time-dependent Schrödinger equation – v5

Project for Computational Physics II, WS19/20 *December 18, 2019*

Calendar

Date	Activity
Nov 19	planned tutorial/lecture
Nov 20	planned tutorial/lecture
Nov 21	tutorial/lecture
Nov 26	no tutorial/lecture
Nov 27	no tutorial/lecture
Nov 28	no tutorial/lecture
Dec 3	planned tutorial/lecture
Dec 4	planned tutorial/lecture
Dec 5	tutorial/lecture
Dec 10	optional tutorial/lecture
Dec 11	optional tutorial/lecture
Dec 12	tutorial/lecture
Dec 17	planned tutorial/lecture
Dec 18	planned tutorial/lecture
Dec 19	**no lecture**
Jan 7	planned tutorial/lecture
Jan 8	planned tutorial/lecture
Jan 8	**deadline** at 23.59

- **This document will be updated. Please check the Moodle page regularly!**
- The grade of this project will contribute to the 30% of the final grade. The grade g will be given on a scale from 0 to 100, and it can be converted to the German academic grading system by using the formula

$$f(g) = 1 + \frac{3}{50}(100 - g) \tag{1}$$

  and by rounding to the closest German grade.
- The list of assessed competences and a description of their indicators can be found in the Moodle page of the course. The weight of each competence in the grading of this project is given in the following table.

Weight	Assessed competence
0%	Physical Transcription
20%	Planning
20%	Implementation
20%	Testing
10%	Running
20%	Numerical Analysis
10%	Visualization + Physical Analysis

- Recommended group size: 3 or 4 students. Please send me an email per group with the list of group members.
- You can ask for feedback from me while you are working on the project (**and you should do it at least once per project**, for instance during the optional exercise session or in my office. We can discuss all aspects of the project: planning choices, code bugs and problems, theoretical and algorithmic aspects. However when you ask for my feedback you must come with clear questions, you must show to me that you have thought about problems.

Contents

1 Overview of goals **3**

2 Overview of main steps **3**

3 Theory introduction: discretization **5**

3.1 Discretization of space and of the wave function 5

3.2 Discretization of derivatives 6

3.3 Boundary conditions 7

3.4 Discretization of the Hamiltonian 7

4 Code module: geometry **8**

5 Code module: Hamiltonian **8**

6 Theory intermezzo: 3 algorithms to solve Schrödinger equation **9**

6.1 Euler method 10

6.2 Unitary Crank-Nicolson method 10

6.3 Strang splitting method 11

7 Code modules: Integrators **12**

8 Some physical observables **13**

9 Choosing parameters with an example **14**

1 Overview of goals

- Design and write a code to solve a discretized version of the time-dependent Schrödinger equation

$$i\hbar\frac{\partial}{\partial t}\psi_t = H\psi_t \tag{2}$$

for the point-particle in D dimensions in external potential. $\psi_t(\mathbf{x})$ is the wave function of the particle at time t, and H is the Hamiltonian

$$H = \frac{\mathbf{p}^2}{2m} + V(\mathbf{x}) = -\frac{\hbar^2}{2m}\boldsymbol{\nabla}^2 + V(\mathbf{x}) \ . \tag{3}$$

In particular, notice that we want to write a code that works for any value of D.

- Use and compare different algorithms to solve the Schrödinger equation.
- Write an extensive set of tests for the implemented algorithms and functions.
- Use this code to visualize the evolution of the wave function of the point particle for a choice of potentials. Possible interesting potentials are: the harmonic oscillator, the potential well, the potential wall, the potential wall with double slits, the Coulomb potential.
- Investigate the dependence of the solution of the Schrödinger equation on the parameters of the discretization (the lattice spacing, the size of the box, the time step).

The project will be split in various steps and you will be guided through these steps, especially at the beginning.

2 Overview of main steps

Discretization of space. We need to discretize the space, i.e. replace the continuum space with a regular finite lattice, with lattice spacing a and N points in each direction (for a total of N^D points).

Definition of geometry. Every point of the lattice is identified by its own coordinates. It is also convenient to assign a unique integer number (*index*) to each point of the lattice. The lattice points are ordered by their index in a sequential manned. There are many ways to indicize the lattice points. The simplest and more intuitive way is the lexicographic ordering (see e.g. Wikipedia). The geometry is defined by the functions that translate indices into coordinates and vice versa. Once the indices are given, one can think the discretized wave function as a one-index array, i.e. a vector, where the index coincides with the index of the lattice points.

Definition of data structure. A discretized wave function is assigned by giving the value of the probability amplitude (which is a complex number) for each point of the lattice $\mathbf{n}$, or which is the same for each value of the index i. Therefore the discretized wave function can be denoted by $\hat{\psi}_i$, which clearly shows that this is a vector. This means that discretized wave function can be represented as one-dimensional arrays, independently of the space dimension D.

Calculation of the Hamiltonian applied to a wave function. Since the discretized wave-function is a vector, the discretized Hamiltonian can be thought mathematically as a matrix. However *we never want to represent it as a two-dimensional array in our code.* As we will discuss, this is dictated both by memory and performance efficiency. We want to write a function that, given $\hat{\psi}$, calculates $\hat{H}\hat{\psi}$. The Hamiltonian has two pieces: the kinetic term $\hat{K}$ and the potential term $\hat{V}$. It is convenient to treat these two terms separately. We want to calculate $\hat{K}\hat{\psi}$ in two ways: directly in coordinate space, or by means of the FFT.

Discretization of the Schrödinger equation. At this point we need to introduce the time variable t. In the continuum, the Schrödinger equation is solve by

$$|\psi_t\rangle = e^{-i\frac{1}{\hbar}Ht}|\psi_0\rangle \ . \tag{4}$$

In the discretized space, this simply becomes

$$\hat{\psi}(t) = e^{-i\frac{\mathcal{E}}{\hbar}\hat{H}t}\hat{\psi}(0) \ , \tag{5}$$

where $e^{-i\frac{\mathcal{E}}{\hbar}\hat{H}t}$ is the exponential of a matrix. It is very expensive to calculate this exponential exactly, therefore we will need to approximate it. We can introduce a time step τ, and discretize the time variable, i.e. we assume $t = \tau r$ where r is an integer. Then we observe that

$$e^{-i\frac{\mathcal{E}}{\hbar}\hat{H}t} = e^{-i\frac{\mathcal{E}\tau}{\hbar}\hat{H}r} = \left[e^{-i\frac{\mathcal{E}\tau}{\hbar}\hat{H}}\right]^r \ . \tag{6}$$

If the parameter $\frac{\mathcal{E}\tau}{\hbar}$ is small enough, the exponential can be approximated by means of various methods. We will consider two methods: the Strang splitting method and the unitary Crank-Nicolson method. We will discuss these in the lectures, however you can read about them in *Numerical Recipes*.

Analysis of the solution of the Schrödinger equation for a couple of potentials. We want to choose some potential, and study a number of properties of the solution of the Schrödinger equation, e.g. probability conservation, energy conservation. We want to understand the effects of discretization, i.e. understand how small a and τ need to be and how large N needs to be in order to get a good approximation of the continuum infinite-volume physics. We want to use the solution to calculate the energy levels by means of the discrete Fourier Transform.

3 Theory introduction: discretization

3.1 Discretization of space and of the wave function

At the moment, we neglect the time variable and we focus only on the space dependence of the wave function. The wave function ψ is a function that associate to each point $\mathbf{x}$ of the space a complex number $\psi(\mathbf{x})$. The point $\mathbf{x}$ is identified by its own coordinates, i.e.

$$\mathbf{x} = (x_1, \ldots, x_D) \ , \tag{7}$$

where x_ℓ is a real number.

In order to be able to represent wave functions on the computer we need to *discretize* the space, i.e. we replace the continuous space with a lattice of point. If a is the lattice spacing, then the points of the lattice are denoted by $a\mathbf{n}$ where $\mathbf{n}$ is an integer vector

$$\mathbf{n} = (n_1, \ldots, n_D) \tag{8}$$

where n_ℓ is an integer.

There are still infinitely many points in the lattice, so this is still not good enough for a computer. The second step is to imagine that the space is not infinite, but it is replaced by a (hyper)cubic box. We can require that x_ℓ is not any real number, but it satisfies

$$0 \leq x_\ell < L \ . \tag{9}$$

where L is the (linear) size of the box. When we discretize the space, we need to assume that L is an integer multiple of a, i.e.

$$L = Na \tag{10}$$

for some natural number N. It follows that the points of the lattice are $a\mathbf{n}$ where $\mathbf{n}$ is a vector

$$\mathbf{n} = (n_1, \ldots, n_D) \quad \text{with} \quad n_\ell = 0, 1, 2, \ldots, N-1 \ . \tag{11}$$

Notice that eventually we will want to take the $a \to 0$ (continuum) limit, and the $L \to \infty$ (infinite-volume) limit. We will discuss this later.

What it is important to notice is that a computer is not able to understand physical units. So we will never be able to represent properly dimensionful quantities in the computer, but only dimensionless combinations. One always needs to use dimensional analysis to understand what are the dimensionless combinations. For instance

$$[x_\ell] = [a] = [L] = \text{length} \ , \tag{12}$$

$$[n_\ell] = \left[\frac{x_\ell}{a}\right] = \frac{\text{length}}{\text{length}} = \text{dimensionless} \ , \tag{13}$$

$$[N] = \left[\frac{L}{a}\right] = \frac{\text{length}}{\text{length}} = \text{dimensionless} \ . \tag{14}$$

Therefore we will never be able to represent a, $\mathbf{x}$ or L in our code, but only $\mathbf{n}$ and N.

Similarly, the wave function has dimensions

$$[\psi(\mathbf{x})] = \text{length}^{-D/2} \ . \tag{15}$$

If you do not know this, you should find why this is the case. We can choose to define the following dimensionless combination

$$\hat{\psi}(\mathbf{n}) = a^{D/2}\psi(a\mathbf{n}) \ . \tag{16}$$

The function $\hat{\psi}(\mathbf{n})$ is dimensionless and it is a function of the dimensionless variable $\mathbf{n}$. Moreover $\mathbf{n}$ is a variable that can take only a finite number of values (how many?).

In the following we will use the following notations interchangeably:

$$\hat{\psi}(\mathbf{n}) = \hat{\psi}_{\mathbf{n}} = \hat{\psi}_{n_1,\ldots,n_D} \ . \tag{17}$$

In particular the last notation shows that, after discretization, $\hat{\psi}$ can be thought as multidimensional array, rather than a function.

3.2 Discretization of derivatives

The kinetic part of the Hamiltonian is written in terms of the nabla operator, i.e. of the partial derivatives with respect to the coordinates. Since we replace the space with a lattice, we also need to replace the derivatives with some discrete approximation. For instance one can observe that

$$\nabla_\ell \psi(\mathbf{x}) = \frac{\partial}{\partial x_\ell}\psi(\mathbf{x}) = \tag{18}$$

$$= \frac{\psi(x_1,\ldots,x_{\ell-1},x_\ell+a,x_{\ell+1}\ldots,x_D)-\psi(x_1,\ldots,x_D)}{a} + O(a^2) = \tag{19}$$

$$= \frac{\psi(\mathbf{x}+a\mathbf{e}_\ell)-\psi(\mathbf{x})}{a} + O(a^2)\ , \tag{20}$$

which is valid in the limit $a \to 0$. The vector $\mathbf{e}_\ell$ is the vector with all zeroes and a one in the ℓ-th component. If $\mathbf{x}$ is a point of the lattice, i.e. it is of the form $\mathbf{x} = a\mathbf{n}$, then one obtains

$$\nabla_\ell \psi(a\mathbf{n}) = \frac{\psi(a(\mathbf{n}+\mathbf{e}_\ell))-\psi(a\mathbf{n})}{a}\ . \tag{21}$$

Notice that the derivative has dimensions:

$$[\nabla_\ell] = \text{length}^{-1}\ . \tag{22}$$

Therefore it makes sense to define the dimensionless discrete derivative $\hat{\nabla}^f$ as

$$\hat{\nabla}^f \hat{\psi}(\mathbf{n}) = \hat{\psi}(\mathbf{n}+\mathbf{e}_\ell) - \hat{\psi}(\mathbf{n})\ . \tag{23}$$

This discrete derivative approximates $a\nabla_\ell$ (which is dimensionless), in fact

$$\hat{\nabla}^f_\ell \hat{\psi}(\mathbf{n}) = \hat{\psi}(\mathbf{n}+\mathbf{e}_\ell) - \hat{\psi}(\mathbf{n}) = a^{3/2}\psi(a(\mathbf{n}+\mathbf{e}_\ell)) - a^{3/2}\psi(a\mathbf{n}) = \tag{24}$$

$$= a^{3/2} a \frac{\psi(a(\mathbf{n}+\mathbf{e}_\ell))-\psi(a\mathbf{n})}{a} \simeq a^{3/2} a \nabla_\ell \psi(a\mathbf{n}) = \widehat{(a\nabla_\ell)}\hat{\psi}(\mathbf{n})\ . \tag{25}$$

The f as a superscript stands for *forward*. One can choose also a *backward* discretization of the derivative, i.e.

$$\hat{\nabla}^b \hat{\psi}(\mathbf{n}) = \hat{\psi}(\mathbf{n}) - \hat{\psi}(\mathbf{n}-\mathbf{e}_\ell)\ , \tag{26}$$

or a symmetric discretization of the derivative

$$\hat{\nabla}^s \hat{\psi}(\mathbf{n}) = \frac{\hat{\psi}(\mathbf{n}+\mathbf{e}_\ell)-\hat{\psi}(\mathbf{n}-\mathbf{e}_\ell)}{2}\ . \tag{27}$$

For the kinetic term of the Hamiltonian we are interested in the Laplacian, i.e. the second derivatives. A commonly used discretization of the Laplacian is obtained by applying the forward derivative first and the backward derivative after

$$\widehat{\boldsymbol{\nabla}^2} = \sum_{\ell=1}^{D} \hat{\nabla}^b_\ell \hat{\nabla}^f_\ell\ . \tag{28}$$

Prove that

$$\widehat{\boldsymbol{\nabla}^2}\hat{\psi}(\mathbf{n}) = \sum_{\ell=1}^{D} \{\hat{\psi}(\mathbf{n}+\mathbf{e}_\ell) - 2\hat{\psi}(\mathbf{n}) + \hat{\psi}(\mathbf{n}-\mathbf{e}_\ell)\}\ . \tag{29}$$

3.3 Boundary conditions

When calculating the action of the discretized Laplacian on the discretized wave function, we need to consider the points $\mathbf{n}+\mathbf{e}_\ell$ and $\mathbf{n}-\mathbf{e}_\ell$. These are called the *nearest neighbours* of $\mathbf{n}$.

Recall that our lattice is finite, i.e. the coordinates n_ℓ take values in $0, 1, \ldots, N-1$. There is a problem of definitions of the nearest neighbours at the boundary of the lattice. For instance, for $D=2$ and $N=8$, we consider the point $\mathbf{n}=(3,7)$. Its nearest neighbour $\mathbf{n}+\mathbf{e}_2=(3,8)$ is a point outside of the lattice, and needs to be properly defined.

We will consider here the case of *periodic boundary conditions*, i.e. we will think the lattice as periodic in all directions. This essentially means that the sum of coordinates should be defined modulo N. In particular, in the example above, $\mathbf{n}+\mathbf{e}_2=(3,0)$, which is a legitimate point of the lattice.

3.4 Discretization of the Hamiltonian

Notice that the Hamiltonian has the dimension of an energy

$$[H] = \text{energy} = \frac{\text{mass} \times \text{length}^2}{\text{time}^2} \ . \tag{30}$$

In order to construct a dimensionless quantity, we need to introduce a constant $\mathcal{E}$ with the dimension of an energy. In principle $\mathcal{E}$ can be arbitrary, however a reasonable choice would be a typical energy scale of the considered system. Then the ration $H/\mathcal{E}$ is dimensionless. We now want to write a discretized version of $H/\mathcal{E}$.

Let us look at the potential $V(\mathbf{x})$ first. As done for the wave function, we need to restrict $\mathbf{x}$ to be a point of the lattice, i.e. $\mathbf{x}=a\mathbf{n}$, and then we need to divide by $\mathcal{E}$ in order to have a dimensionless quantity, i.e.

$$\hat{V}_{\mathbf{n}} = \hat{V}(\mathbf{n}) = \frac{V(a\mathbf{x})}{\mathcal{E}} \ . \tag{31}$$

The the kinetic part of the Hamiltonian is discretized by replacing $\boldsymbol{\nabla}^2 \to a^{-2}\widehat{\boldsymbol{\nabla}^2}$. Therefore the action of the discretized Hamiltonian on the discretized wave function is

$$\hat{H}\hat{\psi}_{\mathbf{n}} = -\frac{\hbar^2}{2m\mathcal{E}a^2}\widehat{\boldsymbol{\nabla}^2}\hat{\psi}_{\mathbf{n}} + \hat{V}_{\mathbf{n}}\hat{\psi}_{\mathbf{n}} \ . \tag{32}$$

It is natural to introduce the parameter

$$\hat{m} = \frac{m\mathcal{E}a^2}{\hbar^2} \ , \tag{33}$$

in terms of which the discretized Hamiltonian is

$$\hat{H}\hat{\psi}_{\mathbf{n}} = -\frac{1}{2\hat{m}}\widehat{\boldsymbol{\nabla}^2}\hat{\psi}_{\mathbf{n}} + \hat{V}_{\mathbf{n}}\hat{\psi}_{\mathbf{n}} \ . \tag{34}$$

The parameter $\hat{m}$ is dimensionless, in fact

$$[\hbar] = \text{energy} \times \text{time} \ , \tag{35}$$

$$[\hat{m}] = \left[\frac{m\mathcal{E}a^2}{\hbar^2}\right] = \frac{\text{mass} \times \text{energy} \times \text{length}^2}{(\text{energy} \times \text{time})^2} = \frac{\text{mass} \times \text{length}^2}{\text{energy} \times \text{time}^2} = \text{dimensionless} \ . \tag{36}$$

4 Code module: geometry

The code **must** be written in a modular form. We want to write a module (or library) that contains all functions and variables that are needed to define the geometry.

This module should contain:

- a function `index2coord` that takes the index of a lattice point and gives back its coordinates;
- a function `coord2index` that takes the coordinates of a lattice point and gives back its index;
- a function `nneighbour` that takes the index of a lattice point and calculates the index of its nearest neighbours with periodic boundary conditions.

Notice that all these functions depend on the parameters N and D. These parameters should be chosen at the beginning of the code and should never change. The geometry module should provide a way to set this variables, and a function that checks that their value has not changed.

The nearest neighbours need to be used in the calculation of the Laplacian. We want this calculation to be as fast as possible. The geometry module should provide a way to calculate all the nearest neighbours once, and to store them in a suitable structure that can be easily accessed by the Laplacian function.

A number of test should be written for every module. For instance, some tests that were suggested in class are (but more can and should be designed)

- Test that `index2coord` and `coord2index` are one the inverse of the other.
- Check that if you calculate the nearest neighbour in direction $+\mathbf{e}_\ell$ and then the nearest neighbour in the direction $-\mathbf{e}_\ell$, you go back to the original point.
- Check that, if you span over all indices, the output of the function `index2coord` spans over all possible coordinates. Also each point is obtained exactly once.

5 Code module: Hamiltonian

We want to write a module (or library) that contains all functions that are needed to calculate the Hamiltonian, and related operators.

This module should contain the following functions, or similar:

- A function `H` that takes the array $\hat{\psi}$ and returns $\hat{H}\hat{\psi}$ as defined in eq. (32). The goal is to get this as fast as possible.
- A number of functions that create the array $\hat{V}$ for different potentials (e.g. harmonic oscillator, Coulomb potential, potential well/wall). These functions need to take the parameters that define the potential as input. Remember that these parameters should be defined only as dimensionless combinations of the physical parameters and the arbitrary scales a and $\mathcal{E}$.
- A function that calculates the average energy of the state $\hat{\psi}$, i.e.

$$\frac{(\hat{\psi}, \hat{H}\hat{\psi})}{(\hat{\psi}, \hat{\psi})} \,. \tag{37}$$

This module should also have a way to set the potential and the parameter $\hat{m}$ which are used by the function H.

Design a number of tests for the `H` function. For instance, you could test that $\hat{H}\hat{\psi}$ is a linear function of $\hat{\psi}$, or the fact that $\hat{H}$ is a hermitean matrix. If you choose $\mathbf{V} = 0$, you should be able to find particular wave functions $\hat{\psi}$ for which the result of $\hat{H}\hat{\psi}$ is analytically known.

The action of the Laplacian or, which is the same, the free Hamiltonian is analytically known on plane waves, i.e. on wave functions of the type

$$\psi_{\mathbf{n}} = e^{i\frac{2\pi}{N}\mathbf{kn}} \ , \tag{38}$$

where $\mathbf{k}$ may be choses as a random vector of integers. Calculate $\widehat{\boldsymbol{\nabla}^2}\psi$ for this particular choce of the wave function, and show that is it proportional to ψ itself (i.e. the plane waves are eigenvectors of the discretized Laplacian). Use this fact to design the analytic test.

Analyse the performance of $\hat{H}$ with a simple potential, i.e. a constant, as a function of N for $D = 1, 2, 3$.

6 Theory intermezzo: 3 algorithms to solve Schrödinger equation

After discretizing space, the Schrödinger equation takes the following form

$$\frac{\partial}{\partial t}\hat{\psi}(t) = -i\frac{\mathcal{E}}{\hbar}\hat{H}\hat{\psi}(t) \ . \tag{39}$$

In this equation, for fixed value of t, $\psi(t)$ is a vector and $\hat{H}$ is a matrix. This equation is solved by

$$\hat{\psi}(t) = e^{-i\frac{\mathcal{E}}{\hbar}t\hat{H}}\hat{\psi}(0) \ , \tag{40}$$

where $\hat{\psi}(0)$ is the initial condition and $e^{-i\frac{\mathcal{E}}{\hbar}t\hat{H}}$ is the exponential of a matrix. Given the matrix M, its exponential can be defined in many equivalent ways, for instance

$$e^M = \sum_{n=0}^{\infty}\frac{M^n}{n!} \ . \tag{41}$$

However the truncation of this series gives a good approximation only it M is a matrix with small entries. In general, one needs to develop better strategies to calculate the exponential.

We introduce a time step τ and we use observe that the Schrödinger equation can be written in the recursive form

$$\hat{\psi}(t+\tau) = e^{-i\frac{\mathcal{E}}{\hbar}(t+\tau)\hat{H}}\hat{\psi}(0) = e^{-i\frac{\mathcal{E}}{\hbar}\tau\hat{H}}e^{-i\frac{\mathcal{E}}{\hbar}t\hat{H}}\hat{\psi}(0) = e^{-i\frac{\mathcal{E}}{\hbar}\tau\hat{H}}\hat{\psi}(t) \ . \tag{42}$$

As we have done in section 3, we write everything in terms of dimensionless quantitities. In particular we introduce the parameter

$$\hat{\tau} = \frac{\mathcal{E}\tau}{\hbar} \tag{43}$$

(check that it is dimensionless), and we measure the time t in units of τ, which means that we replace

$$t \to q\tau \ , \quad \text{with } q = 0, 1, 2, 3, \dots \ , \hat{\psi}(t) \to \hat{\psi}^{(q)} \ . \tag{44}$$

The time-dependent dimensionful wave function is related to the dimensionless one by

$$\psi_{q\tau}(a\mathbf{n}) = \hat{\psi}^{(q)}_{\mathbf{n}} \ . \tag{45}$$

Eq. (42) is written as

$$\hat{\psi}^{(q+1)} = e^{-i\hat{\tau}\hat{H}}\hat{\psi}^{(q)} \ . \tag{46}$$

If $\hat{\tau}$ is small enough, the exponential can be easily approximated in various ways.

6.1 Euler method

The Euler method consists in approximating the exponential in eq. (46) with its first-order Taylor expansion

$$e^{-i\hat{\tau}\hat{H}} \to 1 - i\hat{\tau}\hat{H} \ . \tag{47}$$

This method is very simple to implement, but it is also the least precise. Notice that the error made in a single integration step is

$$e^{-i\hat{\tau}\hat{H}} - [1 - i\hat{\tau}\hat{H}] = O(\hat{\tau}^2) \ . \tag{48}$$

In practice one writes a function that, given $\hat{\psi}^{(q)}$ calculates $\hat{\psi}^{(q+1)}$ with the formula

$$\hat{\psi}^{(q+1)} = \hat{\psi}^{(q)} - i\hat{\tau}\hat{H}\hat{\psi}^{(q)} \ . \tag{49}$$

6.2 Unitary Crank-Nicolson method

The Crank-Nicolson method consists in approximating the exponential in eq. (46) with the following expression

$$e^{-i\hat{\tau}\hat{H}} \to \frac{1 - \frac{i}{2}\hat{\tau}\hat{H}}{1 + \frac{i}{2}\hat{\tau}\hat{H}} \ . \tag{50}$$

If we Taylor expand the r.h.s. in $\hat{\tau}$, we find

$$\frac{1 - \frac{i}{2}\hat{\tau}\hat{H}}{1 + \frac{i}{2}\hat{\tau}\hat{H}} = \left\{1 - \tfrac{i}{2}\hat{\tau}\hat{H}\right\}\left\{1 - \tfrac{i}{2}\hat{\tau}\hat{H} - \tfrac{1}{4}\hat{\tau}^2\hat{H}^2 + \tfrac{i}{8}\hat{\tau}^3\hat{H}^3 + O(\hat{\tau}^4)\right\} = \tag{51}$$

$$= 1 - i\hat{\tau}\hat{H} - \tfrac{1}{2}\hat{\tau}^2\hat{H}^2 + \tfrac{3i}{8}\hat{\tau}^3\hat{H}^3 + O(\hat{\tau}^4) \ . \tag{52}$$

The first three terms coincide with the Taylor expansion of the exponential, which implies

$$e^{-i\hat{\tau}\hat{H}} - \frac{1 - \frac{i}{2}\hat{\tau}\hat{H}}{1 + \frac{i}{2}\hat{\tau}\hat{H}} = O(\hat{\tau}^3) \ . \tag{53}$$

The implemenatation of this algorithm requires the calculation of the inverse of $(1 + \frac{i}{2}\hat{\tau}\hat{H})$. This can be done in sevaral ways. The method proposed here never stores $(1 + \frac{i}{2}\hat{\tau}\hat{H})^{-1}$ as a matrix, but only calculates the inverse multiplied to a vector. Other approaches are possible, especially for lower-dimensional systems. However the proposed method is very general.

First we rewrite

$$\frac{1 - \frac{i}{2}\hat{\tau}\hat{H}}{1 + \frac{i}{2}\hat{\tau}\hat{H}} = \frac{1 - \frac{i}{2}\hat{\tau}\hat{H}}{1 + \frac{i}{2}\hat{\tau}\hat{H}}\frac{1 - \frac{i}{2}\hat{\tau}\hat{H}}{1 - \frac{i}{2}\hat{\tau}\hat{H}} = \frac{\left(1 - \frac{i}{2}\hat{\tau}\hat{H}\right)^2}{1 + \frac{1}{4}\hat{\tau}^2\hat{H}^2} \ . \tag{54}$$

The advantage of this way of writing is that the matrix that we need to invert, i.e. $(1 + \frac{1}{4}\hat{\tau}^2\hat{H}^2)$ is positive, and one can use the so-called *Conjugate Gradient* algorithm.

The calculation is split in the following steps

$$\eta^{(q)} = \left(1 + \tfrac{1}{4}\hat{\tau}^2\hat{H}^2\right)^{-1}\psi^{(q)} \ , \tag{55}$$

$$\psi^{(q+1)} = \left(1 - \tfrac{i}{2}\hat{\tau}\hat{H}\right)^2\eta^{(q)} \ , \tag{56}$$

and the first step is calculated by means of the *Conjugate Gradient* algorithm.

6.3 Strang splitting method

This method is very different from the others. It is based on the decomposition of the Hamiltonian $\hat{H} = \hat{K} + \hat{V}$ in kinetic part

$$\hat{K} = -\frac{1}{2\hat{m}}\widehat{\boldsymbol{\nabla}^2} \tag{57}$$

and potential $\hat{V}$. Then one can use the approximation

$$e^{-i\hat{\tau}\hat{H}} \to e^{-\frac{i}{2}\hat{\tau}\hat{V}} e^{-i\hat{\tau}\hat{K}} e^{-\frac{i}{2}\hat{\tau}\hat{V}} \ . \tag{58}$$

Notice that the two sides are not equal because for matrices in general $e^M e^N \neq e^{M+N}$. In fact, by expanding both sides one finds

$$e^{-i\hat{\tau}\hat{H}} - e^{-\frac{i}{2}\hat{\tau}\hat{V}} e^{-i\hat{\tau}\hat{K}} e^{-\frac{i}{2}\hat{\tau}\hat{V}} = O(\hat{\tau}^3) \ . \tag{59}$$

By using the identities $(e^M)^\dagger = e^{M^\dagger}$ and $(e^M)^{-1} = e^{-M}$, one proves that

$$\left[e^{-\frac{i}{2}\hat{\tau}\hat{V}} e^{-i\hat{\tau}\hat{K}} e^{-\frac{i}{2}\hat{\tau}\hat{V}}\right]^\dagger = e^{\frac{i}{2}\hat{\tau}\hat{V}} e^{i\hat{\tau}\hat{K}} e^{\frac{i}{2}\hat{\tau}\hat{V}} = \left[e^{-\frac{i}{2}\hat{\tau}\hat{V}} e^{-i\hat{\tau}\hat{K}} e^{-\frac{i}{2}\hat{\tau}\hat{V}}\right]^{-1} \ , \tag{60}$$

which is the unitarity condition. The Strang splitting method, as the Crank-Nicolson, is unitary, i.e. it conserves probability.

We want to break down the calculation in the following steps

$$\eta^{(q)} = e^{-\frac{i}{2}\hat{\tau}\hat{V}} \psi^{(q)} \ , \tag{61}$$

$$\chi^{(q)} = e^{-i\hat{\tau}\hat{K}} \eta^{(q)} \ , \tag{62}$$

$$\psi^{(q+1)} = e^{-\frac{i}{2}\hat{\tau}\hat{V}} \chi^{(q)} \ . \tag{63}$$

Notice that the first and third steps are identical. The exponential of the potential can be calculated exactly very easily, since the potential is diagonal in coordintate space. For instance the first step is simply

$$\eta^{(q)}_{\mathbf{n}} = e^{-\frac{i}{2}\hat{\tau}\hat{V}_{\mathbf{n}}} \psi^{(q)}_{\mathbf{n}} \ , \tag{64}$$

for all the lattice point $\mathbf{n}$.

The second step is a bit more complicated because the discrete Laplacian $\widehat{\boldsymbol{\nabla}^2}$ is a non-diagonal matrix. However, as in the continuum, it is diagonalized by going to (discrete) Fourier space. Therefore we introduce the multidimensional DFT of η, defined as

$$\tilde{\eta}^{(q)}_{\mathbf{k}} = \sum_{k_1,\ldots,k_D=0}^{N} e^{-i\frac{2\pi}{N}\mathbf{kn}} \eta^{(q)}_{\mathbf{n}} \ . \tag{65}$$

The inverse of the multidimensional DFT is given by

$$\eta^{(q)}_{\mathbf{n}} = \frac{1}{N^D} \sum_{n_1,\ldots,n_D=0}^{N} e^{i\frac{2\pi}{N}\mathbf{kn}} \tilde{\eta}^{(q)}_{\mathbf{k}} \ . \tag{66}$$

With this representation, the action of the discrete Laplacian can be calculated easily

$$\widehat{\boldsymbol{\nabla}}^2\eta^{(q)}(\mathbf{n}) = \sum_{\ell=1}^{D}\{\eta^{(q)}(\mathbf{n}+\mathbf{e}_\ell) - 2\eta^{(q)}(\mathbf{n}) + \eta^{(q)}(\mathbf{n}-\mathbf{e}_\ell)\} = \tag{67}$$

$$= \frac{1}{N^D}\sum_{n_1,\ldots,n_D=0}^{N-1}\sum_{\ell=1}^{D}\{e^{i\frac{2\pi}{N}\mathbf{k}(\mathbf{n}+\mathbf{e}_\ell)} - 2e^{i\frac{2\pi}{N}\mathbf{kn}} + e^{i\frac{2\pi}{N}\mathbf{k}(\mathbf{n}-\mathbf{e}_\ell)}\}\tilde{\eta}_{\mathbf{k}}^{(q)} = \tag{68}$$

$$= \frac{1}{N^D}\sum_{n_1,\ldots,n_D=0}^{N-1} e^{i\frac{2\pi}{N}\mathbf{kn}}\sum_{\ell=1}^{D}\{e^{i\frac{2\pi}{N}\mathbf{ke}_\ell} - 2 + e^{-i\frac{2\pi}{N}\mathbf{ke}_\ell}\}\tilde{\eta}_{\mathbf{k}}^{(q)} = \tag{69}$$

$$= \frac{1}{N^D}\sum_{n_1,\ldots,n_D=0}^{N-1} e^{i\frac{2\pi}{N}\mathbf{kn}}\sum_{\ell=1}^{D}\{e^{i\frac{2\pi}{N}k_\ell} - 2 + e^{-i\frac{2\pi}{N}k_\ell}\}\tilde{\eta}_{\mathbf{k}}^{(q)} = \tag{70}$$

$$= \frac{1}{N^D}\sum_{n_1,\ldots,n_D=0}^{N-1} e^{i\frac{2\pi}{N}\mathbf{kn}}\sum_{\ell=1}^{D}\{e^{i\frac{\pi}{N}k_\ell} - e^{-i\frac{\pi}{N}k_\ell}\}^2\tilde{\eta}_{\mathbf{k}}^{(q)} = \tag{71}$$

$$= \frac{1}{N^D}\sum_{n_1,\ldots,n_D=0}^{N-1} e^{i\frac{2\pi}{N}\mathbf{kn}}\left\{-4\sum_{\ell=1}^{D}\sin^2\left(\frac{\pi}{N}k_\ell\right)\right\}\tilde{\eta}_{\mathbf{k}}^{(q)} \ . \tag{72}$$

One can write this as

$$\texttt{DFT}\left[\widehat{\boldsymbol{\nabla}}^2\eta^{(q)}\right]_{\mathbf{k}} = \left\{-4\sum_{\ell=1}^{D}\sin^2\left(\frac{\pi}{N}k_\ell\right)\right\}\tilde{\eta}_{\mathbf{k}}^{(q)} \ , \tag{73}$$

i.e. in discrete Fourier space the action of the discrete Laplacian is simply the multiplication by the factor in curly brakets in the above line. The exponential of the kinetic energy is therefore easily calculated in discrete Fourier space

$$\texttt{DFT}\left[e^{-i\hat{\tau}\hat{K}}\eta^{(q)}\eta^{(q)}\right]_{\mathbf{k}} = e^{-i\frac{2\hat{\tau}}{\hat{m}}\sum_{\ell=1}^{D}\sin^2\left(\frac{\pi}{N}k_\ell\right)}\tilde{\eta}_{\mathbf{k}}^{(q)} \ . \tag{74}$$

Therefore the calculation can be split in the following steps

$$\eta_{\mathbf{n}}^{(q)} = e^{-\frac{i}{2}\hat{\tau}\hat{V}_{\mathbf{n}}}\psi_{\mathbf{n}}^{(q)} \ , \tag{75}$$

$$\tilde{\eta}^{(q)} = \texttt{DFT}[\eta^{(q)}] \ , \tag{76}$$

$$\tilde{\chi}_{\mathbf{k}}^{(q)} = e^{-i\frac{2\hat{\tau}}{\hat{m}}\sum_{\ell=1}^{D}\sin^2\left(\frac{\pi}{N}k_\ell\right)}\tilde{\eta}_{\mathbf{k}}^{(q)} \ , \tag{77}$$

$$\chi^{(q)} = \texttt{DFT}^{-1}[\tilde{\chi}^{(q)}] \ , \tag{78}$$

$$\psi_{\mathbf{n}}^{(q+1)} = e^{-\frac{i}{2}\hat{\tau}\hat{V}_{\mathbf{n}}}\chi^{(q)} \ . \tag{79}$$

7 Code modules: Integrators

We want to write three different modules, one per integrator.

The module for the Crank-Nicolson method should include an implementation (written by you) of the Conjugate Gradient algorithm, which you can find on Wikipedia (or if you want more details, on Numerical recipes). You should also think about ways to test the Conjugate Gradient algorithm.

> On Wikipedia, the Conjugate Gradient algorithm is written only for real, symmetric, positive-definite matrices. However the algorithm can be extended to complex, hermitean, positive-definite matrices by replacing all real scalar products $v^t w$ with complex scalar products $v^\dagger w$.

The module for the Strang-splitting method should make use of library FFT algorithms. If you are using C, you can use the GNU Scientific Library.

8 Some physical observables

Once you have chosen the particular form of the discretized potential, an initial condition $\hat{\psi}(0)$, the integratator (Euler, Crank-Nicolson, Strang splitting) and the integration step $\hat{\tau}$, you can solve the Schrödinger equation by applying the integrator several times

$$\hat{\psi}(0) \xrightarrow{\text{integrator}} \hat{\psi}(\hat{\tau}) \xrightarrow{\text{integrator}} \hat{\psi}(2\hat{\tau}) \xrightarrow{\text{integrator}} \hat{\psi}(3\hat{\tau}) \xrightarrow{\text{integrator}} \dots \tag{80}$$

The idea it that you do not want to save the list of all wave-functions in the RAM, because this would generally require too much memory. One can either save them on disk (this may be convenient in C, if you want to visualize them with some different program), or one can calculate some observables *on the fly*, and just replace the wave-function at a given time with the wave-function at the subsequent time.

This is a list of observables that may be interesting to calculate. Some of these observables contain physical information and some of them are useful to understand if the parameters (e.g. lattice spacing, volume) have been chosen wisely or if the chosen algorithm is working properly.

- **Normalization.** This is defined as

  $$\hat{N} = \hat{\psi}^\dagger \psi \ . \tag{81}$$

  If you are using a unitary integrator, then the normalization should be time-independent. Even in this case, because of rounding error, if you run the simulation long enough, you will see that the normalization will deviate from the initial one at some point. When this happens, it means that you have either large time-discretization errors (for a non-unitary integrator) or large rounding errors (for a unitary integrator).

- **Average energy.** The (dimensionless) average energy is given by

  $$\hat{E} = \frac{1}{\hat{N}} \hat{\psi}^\dagger \hat{H} \psi \ . \tag{82}$$

  In the limit $\hat{\tau} \to 0$, the energy should be time-independent. However, because of time-discretization errors, the average energy change over time. If the energy changes too much (i.e. more than your target precision), it means that you should choose a smaller value for $\hat{\tau}$. Physical dimensions are obtained by multiplying by the energy scale, i.e. $E = \mathcal{E}\hat{E}$.

- **Average position.** The definition of the average position is tricky because of the periodic boundary conditions. A reasonable definition may depend on the particular chosen physical system. For instance, consider a central potential (e.g. the harmonic oscillator or the Coulomb potential) defined a such a way that the center is positioned in the lattice point

  $$\mathbf{n}_\mathrm{c} = \left(\tfrac{N}{2}, \tfrac{N}{2}, \dots, \tfrac{N}{2}\right) \ , \tag{83}$$

  with N even. Then it makes sense to define the average (dimensionless) position relative to the center, rather than to the origin, i.e.

  $$\hat{\mathbf{r}} = \frac{1}{\hat{N}} \sum_{\mathbf{n}} \hat{\psi}^*_{\mathbf{n}} (\mathbf{n} - \mathbf{n}_\mathrm{c}) \psi_{\mathbf{n}} \ . \tag{84}$$

  Notice that the particles surely feels the finiteness of the box (i.e. finite-volume effects) if any of the components $\hat{r}_\ell$ is close to 0 or to N. If this happens, you should simulate with a larger value of N. Physical dimensions are obtained by multiplying by the lattice spacing, i.e. $\mathbf{r} = a\hat{\mathbf{r}}$.

- **Position width.** The (dimensionless) width of the wave-function can be defined as

  $$\widehat{\Delta r} = \sqrt{\frac{1}{\hat{N}} \sum_{\mathbf{n}} \hat{\psi}^*_{\mathbf{n}} (\mathbf{n} - \mathbf{n}_\mathrm{c} - \hat{\mathbf{r}})^2 \psi_{\mathbf{n}}} = \sqrt{\frac{1}{\hat{N}} \sum_{\mathbf{n}} \hat{\psi}^*_{\mathbf{n}} (\mathbf{n} - \mathbf{n}_\mathrm{c})^2 \psi_{\mathbf{n}} - \hat{\mathbf{r}}^2} \ . \tag{85}$$

  Notice that the particles surely feels the finiteness of the box (i.e. finite-volume effects) if $\widehat{\Delta r}$ is close to $N/2$. If this happens, you should simulate with a larger value of N. Also, the particles surely feels the lattice discretization (i.e. lattice-spacing effects) if $\widehat{\Delta r}$ is close to 1. If this happens, you should simulate with a smaller value of a. Physical dimensions are obtained by multiplying by the lattice spacing, i.e. $\Delta r = a\widehat{\Delta r}$.

- **Average momentum.** There are many ways to define the momentum in the discretized system. We will see here a way that it is different from the one presented in the lecture. Recall that the momentum operator in quantum mechanics is represented by $-i\hbar\boldsymbol{\nabla}$. In the lattice dicretized system, we can take the forward discretization of the derivative. The dimensionless momentum can be defined as

$$\hat{\mathbf{p}} = \frac{1}{i\hat{N}} \sum_{\mathbf{n}} \hat{\psi}^*_{\mathbf{n}} [\hat{\boldsymbol{\nabla}}^f \hat{\psi}]_{\mathbf{n}} \ . \tag{86}$$

  In components

$$\hat{p}_\ell = \frac{1}{i\hat{N}} \sum_{\mathbf{n}} \hat{\psi}^*_{\mathbf{n}} [\psi_{\mathbf{n}+\hat{\mathbf{e}}_\ell} - \hat{\psi}_{\mathbf{n}}] \ . \tag{87}$$

  This formula can be calculated directly, or also by means of the DFT. In fact one can show that

$$\tilde{\psi}_{\mathbf{k}} = \mathtt{DFT}[\psi]_{\mathbf{k}} \ , \tag{88}$$

$$\hat{p}_\ell = \frac{1}{\hat{N}} \sum_{\mathbf{k}} \tilde{\psi}^*_{\mathbf{k}} e^{i\frac{\pi}{N}k_\ell} 2\sin\left(\frac{\pi}{N}k_\ell\right) \tilde{\psi}_{\mathbf{k}} \ . \tag{89}$$

  The momentum in physical units is obtained as in $\mathbf{p} = \hbar a^{-1}\hat{\mathbf{p}}$. It is easy to prove that $|\hat{p}_\ell| \leq 2$. In fact, lattice-spacing effects are under control only if each component $|\hat{p}_\ell|$ is much smaller than 2. If $|\hat{p}_\ell|$ gets close to 2, then you should choose a smaller value for a.

- **Momentum width.** The (dimensionless) momentum width can be defined as

$$\widehat{\Delta p} = \sqrt{-\frac{1}{\hat{N}} \hat{\psi}^\dagger \widehat{\boldsymbol{\nabla}^2} \psi - \hat{\mathbf{p}}^2} \ . \tag{90}$$

  The particles surely feels the lattice discretization (i.e. lattice-spacing effects) if $\widehat{\Delta p}$ is close or larger than 1. If this happens, you should simulate with a smaller value of a. The momentum width in physical units is obtained as in $\Delta p = \hbar a^{-1} \widehat{\Delta p}$.

9 Choosing parameters with an example

The vibrational modes of the hydrogen molecule can be described in first approximation by a simple one-dimensional harmonic oscillator. The hydrogen atom has mass

$$m_{\mathrm{H}} = 1.7 \times 10^{-27} \text{ kg} \ . \tag{91}$$

At equilibrium, the two hydrogen nuclei have distance of about $\bar{r} = 1.2 \times 10^{-10}$ m (this is the so-called Van Der Waals radius of the hydrogen molecule).

Let x be a one-dimensional coordinate along the axis which goes through the two hydorgen nuclei. We want to describe small oscillations of the hydrogen atoms close to the equilibrium position, which we choose to be at $x = 0$. The Hamiltonian discribing the vibrations of the hydrogen molecule (in s-wave) is given by

$$H = \frac{p^2}{m_{\mathrm{H}}} + \frac{1}{4} m_{\mathrm{H}} \omega^2 x^2 \ , \tag{92}$$

where the harmonic-oscillator frequency is experimentally determined to be

$$\omega = 8.3 \times 10^{14} \text{ Hz} \ . \tag{93}$$

We want to answer the following question: is it possible to guess which values of a, L, τ and $\mathcal{E}$ are natural to choose in this problem? Let us recall the main relations involving these parameters.

The number of points is given by

$$N = \frac{L}{a} \ . \tag{94}$$

The integration (dimensionless) time-step is given by

$$\hat{\tau} = \frac{\mathcal{E}\tau}{\hbar} \ . \tag{95}$$

The discretized (dimensionless) Hamiltonian is given by

$$\hat{H} = \left[\frac{H}{\mathcal{E}}\right]_{\text{discretized}} = -\frac{\hbar^2}{m_{\text{H}}\mathcal{E}a^2}\widehat{\nabla^2} + \frac{m_{\text{H}}\omega^2 a^2}{4\mathcal{E}}\left(n - \frac{N}{2}\right)^2 = -\frac{1}{2\hat{m}}\widehat{\nabla^2} + \frac{1}{2}\hat{\kappa}\left(n - \frac{N}{2}\right)^2 \ , \tag{96}$$

where it is convenient to choose the center of the harmonic potential in $n = N/2$, and the following dimensionless parameters

$$\hat{m} = \frac{m_{\text{H}}\mathcal{E}a^2}{2\hbar^2} \ , \tag{97}$$

$$\hat{\kappa} = \frac{m_{\text{H}}\omega^2 a^2}{2\mathcal{E}} = \hat{m}\frac{\hbar^2\omega^2}{\mathcal{E}^2} \ . \tag{98}$$

Notice that these parameters play different roles. In order to recover the continuum physics, we need to take the $a \to 0$ limit; in order to recover the infinite-volume physics, we need to take the $L \to \infty$ limit; in order the continuum-time physics, we need to take the $\tau \to 0$ limit. On the other hand the parameter $\mathcal{E}$ is arbitrary, it sets only the unit of measure of the energy, and can be chosen once and for all. It usually makes sense to try to guess the typical energy scale of the problem. In the case of the harmonic oscillator, this is particularly simple since one can simply choose

$$\mathcal{E} = \hbar\omega = 8.8 \times 10^{-20} \text{ J} \ . \tag{99}$$

A signal of the fact that this choice is particularly convenient, comes from the observation that

$$\hat{\kappa} = \hat{m} = \frac{m_{\text{H}}\omega a^2}{2\hbar} \ , \tag{100}$$

and the discretized Hamiltonian can be expressed in terms of a single parameter (this is not always possible). Notice that (having fixed $\mathcal{E}$ once and for all), assigning a value for the dimensionless parameter $\hat{m}$ is equivalent to assigning a value for the lattice spacing a. In particular sending $a \to 0$ while keeping all physical parameters fixed is equivalent to sending $\hat{m} \to 0$.

Let us look now at the other parameters. The problem has a natural time scale which is given by the period of vibration, i.e. $T = \frac{2\pi}{\omega}$. In order to describe those oscillations properly, one should require at least

$$\tau \ll T = \frac{2\pi}{\omega} \ . \tag{101}$$

In fact, typically these problems have higher-frequancy modes (i.e. modes with smaller momentum). If one wants to make sure to describe correctly all modes with period e.g. $T/10$, then one should require

$$\tau \ll \frac{T}{10} = \frac{\pi}{5\omega} \ . \tag{102}$$

Recalling the definition of $\hat{\tau}$ (which is the only paramters that enters in our simulations), this implies

$$\hat{\tau} = \frac{\mathcal{E}\tau}{\hbar} = \omega\tau \ll \frac{\pi}{5} \simeq 0.6 \ . \tag{103}$$

This inequality gives you an idea of the values that you should choose for $\hat{\tau}$. Eventually you should decrease $\hat{\tau}$ up to a point at which your result do not depend on it any longer.

The system also has a natural lenght scale, which is given by the width of the ground-state wave function, which is given by

$$\Delta x_0 = \sqrt{\frac{\hbar}{m_{\rm H}\omega}} = 8.6 \times 10^{-12} \text{ m} \ . \tag{104}$$

Notice that the width in units of lattice spacings is given by

$$\frac{\Delta x_0}{a} = \sqrt{\frac{\hbar}{m_{\rm H}\omega a^2}} = (2\hat{m})^{-1/2} \ . \tag{105}$$

Excited states are typically broader, however they are oscillating. If we want to be able to describe accurately excited states, then we need the lattice spacing to be smaller than the typical oscillation, and we need the volume to be larger than the width of the excited state. For instance, if we want to be able to describe accurately states with oscillation lenght in the range $[\Delta x_0/5, \Delta x_0]$, and with a total width in the range $[\Delta x_0, 5\Delta x_0]$, we need to require

$$a \ll \frac{\Delta x_0}{5} < 5\Delta x_0 \ll L \ , \tag{106}$$

which is equivalent to

$$1 \ll \frac{1}{5}(2\hat{m})^{-1/2} < 5(2\hat{m})^{-1/2} \ll N \ . \tag{107}$$

This inequality gives an idea of what values for $\hat{m}$ and N are needed to keep the lattice-spacing and finite-volume effects under control. Notice that values for $\hat{m}$ and N translate into values for a and L.

When running a simulation of the time evolution, one should also choose an initial condition for the wave-function. For instance one can use a Gaussian wave packet with width equal to σ. In order to avoid lattice-spacing and finite-volume effects, one should also choose $a \ll \sigma \ll L$.