

MATHEMATISCHES INSTITUT  
DER LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN



**Diplomarbeit**

**Numerical investigations of the long time  
evolution of the Schrödinger equation**

vorgelegt von

Schekeb Sarwari

---

Betreuer Prof. László Erdős Ph.D

Abgabetermin 15. Mai 2009

---

## **Declaration of Authorship**

I certify that the work presented here is, to the best of my knowledge and belief, original and the result of my own investigations, except as acknowledged, and has not been submitted, either in part or whole, for a degree at the LMU or any other university.

München, 15.Mai.2009

---

## Abstract

The time evolution of the Schrödinger equation is subject in many areas such as physics, mathematics, chemistry and optics. Since the time dependent Schrödinger equation can not be solved or it is too expensive to solve for many potentials, the need for numerical approximation techniques arises. From the analytical spadework we know which state variables are conserved. Numerical integrators must either conserve or, in a good manner, approximate the constraints given by the physical system. We investigate the long time evolution of the Schrödinger equation computed by various numerical methods under the condition that our methods conserve the  $l^2$ -norm. We see that the celebrated symmetric operator splitting method does not conserve the energy of the physical system, but the energy related error remains small for, at least, differentiable potentials. It turns out that a good integrator for the short time evolution of the Schrödinger equation provides the method of scaling and squaring using diagonal Padé approximants.

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	Motivation . . . . .	6
1.2	Consistency, stability and convergence . . . . .	8
1.3	Constraints . . . . .	10
<b>2</b>	<b>Operator Splitting</b>	<b>11</b>
2.1	Introduction . . . . .	11
2.2	Symmetric exponential operator splitting . . . . .	13
2.3	Stability and energy conservation . . . . .	19
<b>3</b>	<b>The implicit Crank-Nicholson Method</b>	<b>21</b>
3.1	Construction . . . . .	21
3.2	Stability . . . . .	23
<b>4</b>	<b>Method of Scaling and Squaring</b>	<b>24</b>
<b>5</b>	<b>Testing</b>	<b>28</b>
5.1	A free particle . . . . .	28
5.2	Coherent states of the harmonic Oscillator . . . . .	31
5.3	A random potential . . . . .	33
5.4	A Mexican hat potential . . . . .	36
5.5	An Oscillating potential . . . . .	37
<b>6</b>	<b>Wigner quasi probability distribution</b>	<b>40</b>
<b>7</b>	<b>Conclusion</b>	<b>41</b>
<b>8</b>	<b>Acknowledgment</b>	<b>42</b>
<b>9</b>	<b>Appendix</b>	<b>43</b>
9.1	Dft . . . . .	43
9.2	Quadrature formulae . . . . .	44

# List of Figures

5.1	The initial condition $\psi(x, 0)$ . . . . .	30
5.2	Fourier transform of $\psi(x, 0)$ . . . . .	30
5.3	Error $d(t)$ for $\Phi_{CN}$ , $\Phi_{Spl}$ and $\Phi_{Pad}$ . . . . .	30
5.4	Energy $E(\tau)$ vs. time for $\Phi_{CN}$ , $\Phi_{pad}$ and $\Phi_{Spl}$ . . . . .	30
5.5	$P(x, 0)$ and $V(x)$ . . . . .	32
5.6	$d(t)$ vs. $t$ for the three methods . . . . .	32
5.7	Error function $d(t)$ in case of $\Phi_{spl}(\tau)$ . . . . .	33
5.8	Energy vs. time for the three methods . . . . .	33
5.9	$V(x)$ with $h = \pi/50$ . . . . .	34
5.10	$E(\tau_1)$ and $E(\tau_2)$ . . . . .	34
5.11	$e(t)$ for $T = 600$ , $\tau_1 = 1/10$ and $\tau_2 = 1/20$ . . . . .	34
5.12	$e(t)$ for $T = 3000$ , $\tau_1 = 1/20$ and $\tau_2 = 1/40$ . . . . .	34
5.13	$\psi(x, T = 300)$ . . . . .	35
5.14	$E(t)$ comp. by $\Phi_{CN}(\tau)$ and $\Phi_{\tau}$ . . . . .	35
5.15	$e(t)$ comp. with $\Phi_{CN}(\tau)$ . . . . .	35
5.16	$e(t)$ comp. with $\Phi_{Sas}(\tau)$ . . . . .	35
5.17	Mexican hat potential . . . . .	36
5.18	$\psi(x, T = 300)$ . . . . .	36
5.19	$E(t)$ computed with $\Phi_{CN}(\tau)$ , $\Phi_{Sas}(\tau)$ and $\Phi_{Spl}(\tau)$ . . . . .	36
5.20	$e(t)$ computed with $\Phi_{Spl}(\tau)$ . . . . .	36
5.21	$e(t)$ computed with $\Phi_{Spl}(\tau)$ . . . . .	37
5.22	$e(t)$ computed with $\Phi_{Sas}(\tau)$ . . . . .	37
5.23	Potential $V(x)$ . . . . .	37
5.24	$\psi(x, T = 300)$ . . . . .	37
5.25	$E(\tau)$ computed with $\Phi_{CN}(\tau)$ , $\Phi_{Sas}(\tau)$ and $\Phi_{Spl}(\tau)$ . . . . .	38
5.26	$e(t)$ computed with $\Phi_{Spl}(\tau)$ . . . . .	38
5.27	$e(t)$ computed with $\Phi_{CN}(\tau)$ . . . . .	38
5.28	$e(t)$ computed with $\Phi_{Sas}(\tau)$ . . . . .	38

# 1 Introduction

## 1.1 Motivation

Many problems of modern physics and mathematics are described by ordinary differential equations or partial differential equations. If there is no closed form of the solution, or it is too expensive to calculate, the need for good approximation techniques arises. What constitutes a good approximation is that it inherits properties of the underlying physical system. In the following, our physical system is described by the Schrödinger equation. We choose a finite domain, and restrict a particle to it. This is the often used 'particle in a box' scenario. Then we discretize the position space and the momentum space to get discrete analogs of the observables. Since we are interested in the long time evolution of the Schrödinger equation we have to choose our position space big enough. The goal is to find a numerical scheme that is consistent with the Schrödinger equation, stable and accurate, which conserves both, the  $l^2$ -norm and the energy. We will see that there exists such methods, which are improper for the long time evolution of the Schrödinger equation.

The time dependent Schrödinger equation is a parabolic linear partial differential equation of second order. For a particle of mass  $m$  in a potential  $V(x)$  it reads

$$\begin{aligned} i\hbar \frac{\partial}{\partial t} \psi(x, t) &= H\psi(x, t) \\ \psi(x, t = t_0) &= \psi^0(x) \end{aligned} \tag{1.1}$$

where  $H = -\frac{\hbar^2}{2m}\Delta + V$  is the time-independent Hamiltonian,  $\psi^0(x)$  a given initial condition and  $V(x)$  acts on  $\psi(x, t)$  by multiplication. We may assume  $\hbar=1$ ,  $m = 1$ . Then, a formal solution of the Schrödinger equation (1.1) is given by

$$\psi(x, t) = e^{-itH}\psi^0(x) \tag{1.2}$$

Since for many potentials  $V(x)$  the exact time evolution of (1.2) cannot be solved, one needs reliable numerical methods that approximate solutions of the Schrödinger equation.

For numerical purposes we have to restrict  $x$  and  $t$  to finite domains. Therefore let  $x \in X = [x_0, L]$  and  $t \in [t_0, T]$  for some  $x_0, t_0, L, T \in \mathbb{R}$  such that  $x_0 < L, t_0 < T$  and define the equidistant grid

$$G_h = \{x_0 + jh | 0 \leq j \leq m\}$$

and

$$T_\tau = \{t_0 + k\tau | 0 \leq k \leq n\}$$

for a given small  $h > 0$ ,  $\tau > 0$  and  $n \in \mathbb{N}, m \in \mathbb{N}$ . We write  $x_j := x_0 + jh$  and  $t_k := t_0 + k\tau$  for  $1 \leq j \leq m$  and  $1 \leq k \leq n$ . For the sake of convenience we may assume  $x_0 = -L$  and introduce periodic boundary conditions, i.e.  $\psi(-L, \cdot) = \psi(L, \cdot)$ . Then for any fixed  $k$ , the wave function is represented by an array of its values on the grid  $G_h$ , i.e.

$$\psi_j^k := \psi(x_j, t_k) \quad 0 \leq j \leq m$$

More precisely, for any fixed  $k$ , we have the map  $\psi : G_h \mapsto l^2(I)$  where  $I \subset \mathbb{Z}$  is finite. Let  $l^2(I)$  be equipped with the usual scalar product

$$(x, y) = \sum_{j=1}^m x_j y_j^*,$$

where asterix denotes the complex conjugate. Then the  $l^2$ -norm is denoted by

$$\|x\|_2 = \sqrt{(x, x)}.$$

Unless otherwise stated  $\|\cdot\| = \|\cdot\|_2$  denotes the  $l^2$ -norm in the following. Additionally, we define the grid dependent norm

$$\|v\|_h = \left( h \sum_{j=1}^m |v_j|^2 \right)^{\frac{1}{2}}$$

which allows us easy passage from vector to function norms.

Now we define the discrete differential operator

$$\frac{\partial^2}{\partial x^2} \psi(x_j, t_k) = \frac{\psi(x_{j-1}, t_k) - 2\psi(x_j, t_k) + \psi(x_{j+1}, t_k)}{h^2} + \mathcal{O}(h^2)$$

according to the five point difference star. Thus in the standard basis of the  $\mathbb{R}^N$ , the matrix components of the negative laplacian are given by

$$H_0(i, j) = \begin{cases} 1 & i = j \\ -1/2 & |i - j| = 1 \\ 0 & \text{else} \end{cases}$$

It is a tridiagonal matrix with eigenvalues

$$\lambda_k = -2\sin^2 \left( \frac{k\pi}{2(m+1)} \right) \quad 1 \leq k \leq m$$

where  $m$  is the matrix size.[1]

A discrete version of the multiplication operator  $V(x)$  is given by

$$V_0(x_j) = V(x_j) \quad 1 \leq j \leq m.$$

In the following we omit the index of the Potential and write  $H = H_0 + V$  for the discrete Hamiltonian and  $H = -\frac{1}{2}\Delta + V$  for the continuous Hamiltonian respectively. If nothing stated it will be clear from the context which type of Hamiltonian we use. With  $H = H_0 + V$  we get a discrete version of (1.1)

$$\begin{aligned} i \frac{\partial \psi}{\partial t}(x_j, t_k) &= (H\psi^k)_j \\ \psi(x_j, t_0) &= \psi^0(x_j) \end{aligned}$$

which is a system of ordinary differential equations. Formal integration yields,

$$\psi_j^k = \psi(x_j, t_k) = e^{-ik\tau H} \psi^0(x_j). \quad (1.3)$$

Since we are interested in the long time evolution of the Schrödinger equation, we mostly suppress the index of the spatial variable and seek for schemes  $\Phi$  such that

$$\psi^k = \Phi \psi^0. \quad (1.4)$$

Different approaches have been made to find stable, accurate and fast methods solving the Schrödinger equation. One way dealing with the problem is direct approximation of the time evolution operator  $e^{-ik\tau H}$ . The resulting methods are iterative ones. Another way to compute  $\psi(x, t)$  for various  $t$  is to use the right hand side of equation (1.1) as a tangential slope of the solution at the point  $(x, t)$ . Attempts in that direction leads to so called Runge-Kutta methods. In order to get a better insight of the methods we introduce some terminology.

## 1.2 Consistency, stability and convergence

Let  $L(\phi) = 0$  be the PDE in its implicit form,  $F(\psi) = 0$  be the associated finite difference approximation. If the exact solution  $\psi$  of the difference equation is replaced by the exact solution  $\phi$  of the PDE, the truncation error at the grid point  $(jh, k\tau)$  is given as  $T_{jk}(\phi) = F(\phi_j^k) - L(\phi_j^k) = F(\phi_j^k)$  since  $L(\phi_j^k) = 0$ .

### Definition 1.

The local truncation error at time  $t_k$  is the vector  $\sigma^k(h, \tau) = (T_{jk}(\phi))_{1 \leq j \leq m}$  where  $\phi$  represents the exact solution of the PDE.

**Definition 2.**

A difference equation is called consistent with the differential equation if

$$\lim_{\substack{h \rightarrow 0 \\ \tau \rightarrow 0}} (\|\sigma^k(h, \tau)\|) = 0 \quad \forall k \in \mathbb{N} \text{ with } t_0 + k\tau \leq T$$

A difference equation is consistent of order  $p$  with the differential equation if for a  $k \in \mathbb{N}$  with  $t_0 + k\tau \leq T$  it is valid that

$$\|\sigma^k(h, \tau)\| = \mathcal{O}(\tau^{p+1}) \quad \text{as } \tau \rightarrow 0.$$

Notice that the order of consistency is one less than the local error. That is, because accumulation of the errors leads to a global error of order  $\mathcal{O}(\tau^p)$  which is used as measure for the quality of an algorithm. Also note that we have only defined consistency with respect to the temporal variable. That has two reasons. At first the temporal variable  $\tau$  constitutes the order of energy conservation and therefore, if a method has a high order of consistency it may also have a good energy conservation character. Secondly, the laplacian is discretized in the same manner throughout this work, and therefore any method that utilizes  $H_0$  is of order two accurate with respect to the spatial variable. We should also note that

$$\sigma^k(h, \tau) = \|\phi^{k+1} - \Phi(\tau)\phi^k\|.$$

To define convergence, consider  $\phi(x, t)$  to represent the exact solution of the partial differential equation and  $\psi(x, t)$  to represent the exact solution of the adequate finite difference equation. Let the vectors  $\phi^k = \phi(\cdot, t_k)$  and  $\psi^k = \psi(\cdot, t_k)$  represent the components of  $\phi$  and  $\psi$  respectively at the time level  $t_k = t_0 + k\tau$ . Define the discretisation error as  $e_k = \phi^k - \psi^k$ .

**Definition 3.**

A finite difference equation is said to be convergent in the norm  $\|\cdot\|_h$  when

$$\lim_{h \rightarrow 0} (\max_{\substack{k \\ k\tau \leq T}} \|e_k\|_h) = 0$$

for every initial condition, for every  $T > 0$  and for  $\frac{\tau}{h^2}$  constant as  $h \rightarrow 0$ .

The essential idea behind the concept of stability is that the numerical process should not cause any small perturbations introduced through rounding which grow and ultimately dominate the solution.

**Definition 4.** (*Lax-Richtmyer stability*)

A method is stable if for every  $T > 0$  there exists a constant  $C(T) \geq 0$  independent of  $(h, \tau)$  such that

$$\|\psi^k\| \leq C(T) \quad k = 0, 1, \dots, T/\tau \quad \text{and} \quad h \rightarrow 0, \tau \rightarrow 0$$

One of the fundamental theorems in numerical analysis is due to Peter Lax.

**Theorem 1.** (*Lax equivalence Theorem*)[6] *Given a well posed linear initial value problem and a finite difference approximation to it that satisfies the consistency condition. Then stability is a necessary and sufficient condition for convergence.*

### 1.3 Constraints

The statistical interpretation of wavepackets in Quantum mechanics provides that  $\int_{\mathbb{R}} |\psi|^2 = 1$ , which states that the probability of finding the particle anywhere on the one dimensional universe, i.e. the real line, must be unity. As a minimum request on a numerical scheme we demand the conservation of this property, which means that given a numerical solution  $\psi^k$  of equation (1.1) at a certain time level  $t_k$  then  $\|\psi^k\|_h = 1$  for all  $k \in \mathbb{N}$  with  $k\tau \leq T$ . Moreover, the expectation value of the total energy, represented by the average value of the operator  $\hbar \frac{\partial}{\partial t}$  is constant,

$$(\phi, \hbar \frac{\partial}{\partial t} \phi) = (\phi, H\phi) = \int_{\mathbb{R}} \phi(H\phi)^* dx = const$$

for all times  $t$ , that is, energy is a conserved quantity of the physical system.

We are interested in numerical schemes that conserve this quantity, i.e.

$$(\psi^k, H\psi^k) = h \sum_{j=1}^m \psi_j^k (H\psi_j^k)^* = const \quad \text{for all } k \in \mathbb{N} \text{ such that } k\tau \leq T$$

**Definition 5.** Let  $\psi^k$  be a numerical solution of the Schrödinger equation at time  $t = t_0 + k\tau$ . We say that a numerical method is energy conserving if

$$(\psi^k, H\psi^k) = (\psi^0, H\psi^0) \text{ is valid for all } k \in \mathbb{N} \text{ such that } k\tau \leq T \quad (1.5)$$

In order to be able to distinguish between energy conserving schemes we define the magnitude of energy conservation.

**Definition 6.** Let  $\psi^k$  be a numerical solution of the Schrödinger equation at time  $t_k = t_0 + k\tau$ . We say that a numerical method is energy conserving of magnitude  $p$  if

$$|(\phi^k, \hat{H}\phi^k) - (\psi^k, H\psi^k)| = \mathcal{O}(\tau^p) \quad \text{as } \tau \rightarrow 0$$

is valid for all  $k \in \mathbb{N}$  such that  $t_0 + k\tau \leq T$ .

where  $\hat{H}\psi = (-\frac{1}{2}\Delta + V)\psi$ . On the basis of the last definition one can classify numerical schemes by means of their energy conservation magnitude. Since we are only interested in methods that conserve energy in sense of definition 5, we refer methods as not energy conserving if condition (1.5) is violated.

## 2 Operator Splitting

The Strang splitting, also known as operator splitting method, is one of the fast methods computing the solution of the time dependent Schrödinger equation. Because of its property to conserve the  $l^2$ -norm of the solution it is very attractive in many applications. This class of methods has the advantage that partial derivatives with respect to the spatial variable in positions space can be computed via multiplication in momentum space. Therefore it avoids the approximation of the partial derivatives with respect to the spatial variable.

### 2.1 Introduction

As already seen, the formal solution of the Schrödinger equation is given by

$$\psi(x, t) = e^{-i(t-t_0)H}\psi(x, t_0)$$

Where  $H = -\frac{1}{2}\Delta + V$ . Let  $U(t, t_0) = e^{-i(t-t_0)H}$  be the time-evolution propagator and define for  $k \in \mathbb{N}$  the short time evolution propagator  $U(\tau) = U(t_0 + k\tau, t_0 + (k-1)\tau)$ . Clearly,  $U(\tau) = e^{-i\tau H}$ . Then for  $t = t_0 + k\tau$

$$U(t, t_0) = \prod_{s=1}^k U(\tau) \quad (2.1)$$

Using the separated structure of the Hamiltonian  $H = (-\frac{1}{2}\Delta + V)$  by splitting the short time evolution propagator into a kinetic and a potential operator, i.e.

$$U(\tau) \approx e^{i\frac{\tau}{2}\Delta}e^{-i\tau V}$$

and substituting that back in (2.1) provides an approximation scheme conserving the  $l_2$ -norm of the approximated solution at any time level. For a given  $t = t_0 + k\tau$  the approximation scheme  $\Phi(\tau)$  is given by

$$\psi^{k+1} = \Phi(\tau)\psi^k = \left( \prod_{s=1}^k \Phi(\tau) \right) \psi^0$$

with  $\psi^0 = \psi(\cdot, t_0)$ ,  $\psi^k = \psi(\cdot, t_k)$  and  $\Phi(\tau) = e^{i\frac{\tau}{2}\Delta}e^{-i\tau V}$ . If we set  $t = t_{k+1}$  and substitute  $u(x, t) = \Phi(\tau)\psi^k$  then

$$u(x, t) = e^{i\frac{\tau}{2}\Delta}v(x, t) \quad (2.2)$$

with  $v(x, t) = e^{-i\tau V}\psi^k$  and notice that (2.2) is the solution of the equation

$$\begin{aligned}\frac{\partial}{\partial s}u(x, s) &= i\frac{1}{2}\frac{\partial^2}{\partial x^2}u(x, s) \\ u(x, t_0) &= v(x, t)\end{aligned}\tag{2.3}$$

Fourier transform with respect to  $x$  on both sides yields

$$\hat{u}(\xi, t) = -i\frac{\xi^2}{2}\hat{u}(\xi, t)\tag{2.4}$$

$$\hat{u}(\xi, t_0) = \hat{v}(\xi, t)\tag{2.5}$$

with the solution

$$\hat{u}(\xi, \tau) = e^{-i\frac{\tau}{2}\xi^2}\hat{u}(\xi, t_0)$$

Therefore,

$$\Phi(\tau)\psi(x, t_k) = F^{-1}\left(e^{-i\tau\xi^2}\left(F\left(e^{-i\tau\hat{V}}\psi(x, t_k)\right)\right)\right),$$

where  $F$  and  $F^{-1}$  denotes the (continuous) Fourier transform and its inverse respectively.

So far we have not discretized the spatial domain. In doing that, how it is described above,  $m$  discrete points in position space yield  $m$  discrete points in momentum space. Therefore,

**Definition 7.** *The discrete Fourier transform (DFT) of a vector  $v$  is given by*

$$\hat{v}_{\xi+1} = (\mathcal{F}v)_{\xi+1} = \sum_{j=0}^{N-1} v_{j+1}e^{-2\pi ij\xi/N}$$

*The inverse transform is*

$$v_{j+1} = (\mathcal{F}^{-1}v)_{j+1} = \frac{1}{N}\sum_{\xi=0}^{N-1} \hat{v}_{\xi+1}e^{2\pi ij\xi/N}$$

where  $i = \sqrt{-1}$  and  $\xi = 0, \dots, N-1$ .

Defining the exponential of an bounded operator  $A$  via

$$e^A = I + \sum_{n=1}^{\infty} \frac{1}{n!}A^n$$

the Baker-Campbell-Hausdorff Formula

$$e^{tA}e^{tB} = e^{t(A+B)+t^2\frac{1}{2}[A,B]+t^3\frac{1}{12}([A,[A,B]]+[B,[A,B]])\dots}$$

together with the series expansion of the matrix exponential on both sides yields

$$\|U(\tau)\phi - \Phi(\tau)\phi\| = \mathcal{O}(\tau^2) \quad \text{as } \tau \rightarrow 0.$$

where  $\phi$  is the exact solution of the Schrödinger equation and  $[A, B] := AB - BA$  denotes the commutator of the operators  $A$  and  $B$ . Thus the Strang Splitting is an approximation scheme of order one.

The big drawback of this method is the low order of accuracy. Strang suggested two further splitting methods of order two. The first one uses that  $[A, B] = -[B, A]$  and is given by

$$e^{i\tau(A+B)} = \frac{1}{2}(e^{i\tau A}e^{i\tau B} + e^{i\tau B}e^{i\tau A}) + \mathcal{O}(\tau^3)$$

The second one will be discussed in detail next.

## 2.2 Symmetric exponential operator splitting

For  $H = (H_0 + V)$  the symmetric exponential operator splitting is given by the operator

$$\Phi(\tau) = e^{\frac{-i\tau}{2}V} e^{-i\tau H_0} e^{\frac{-i\tau}{2}V} \quad (2.6)$$

The last exponential can be computed together with the first one, and these operators can be put together exactly for adjacent timesteps. If we switch  $H_0$  and  $V$  in 2.2 we get an equivalent algorithm. However, that one would use four Fourier transforms instead of two.

In order to compute the local error we substitute  $t = -i\tau$ , then for  $H = (H_0 + V)$  the short time evolution propagator is given by  $U(\tau) = e^{t(H_0+V)}$ .

$$\begin{aligned} e^{tH_0} e^{\frac{t}{2}V} &= \left(1 + tH_0 + \frac{t^2}{2}H_0^2\right) \left(1 + \frac{t}{2}V + \frac{t^2}{8}V^2\right) + \mathcal{O}(\tau^3) \\ &= \left(1 + \frac{t}{2}V + \frac{t^2}{8}V^2 + tH_0 + \frac{t^2}{2}H_0V + \frac{t^2}{2}H_0^2\right) + \mathcal{O}(\tau^3). \end{aligned}$$

Therefore,

$$\begin{aligned} \Phi(\tau) &= \left(1 + \frac{t}{2}V + tH_0 + \frac{t}{2}V\right) \\ &\quad + \left(\frac{t^2}{8}V^2 + \frac{t^2}{2}H_0V + \frac{t^2}{2}H_0^2 + \frac{t^2}{4}V^2 + \frac{t^2}{2}VH_0 + \frac{t^2}{8}V^2\right) + \mathcal{O}(\tau^3) \\ &= \left(1 + t(H_0 + V) + \frac{t^2}{2}(H_0^2 + VH_0 + H_0V + V^2)\right) + \mathcal{O}(\tau^3) \\ &= \left(1 + t(H_0 + V) + \frac{t^2}{2}(H_0 + V)^2\right) + \mathcal{O}(\tau^3) \end{aligned}$$

Together with the Taylor series of  $U(\tau) = e^{t(H_0+V)}$  it follows

$$\|U(\tau) - \Phi(\tau)\| = \mathcal{O}(\tau^3) \quad \text{as } \tau \rightarrow 0$$

Which yields that the symmetric exponential operator splitting is a scheme of order two. The convergence of exponential operator splitting methods is assured by the Trotter product formula

**Lemma 2.** (*Trotter Product Formula*) *For two bounded linear operators  $A, B$  it is valid that*

$$\lim_{n \rightarrow \infty} (e^{-\frac{it}{n\tau}A} e^{-\frac{it}{n\tau}B})^n = e^{-\frac{it}{\tau}(A+B)}$$

*Proof.* For matrices, the statement follows with help of the exponential series

$$\begin{aligned} \left( e^{\frac{it}{n\tau}A} e^{\frac{it}{n\tau}B} \right) &= \left( \left( 1 + \frac{it}{n\tau}A \right) \left( 1 + \frac{it}{n\tau}B \right) + \mathcal{O}\left(\frac{1}{n^2}\right) \right)^n \\ &= \left( \left( 1 + \frac{it}{n\tau}(A+B) \right) + \mathcal{O}\left(\frac{1}{n^2}\right) \right)^n \\ &= \left( 1 + \frac{it}{n\tau}(A+B) \right)^n + \mathcal{O}\left(\frac{1}{n}\right) \\ &= \left( e^{\frac{it}{n\tau}(A+B)} \right)^n + \mathcal{O}\left(\frac{1}{n}\right) \end{aligned}$$

for  $n \rightarrow \infty$  the error term vanishes. For bounded linear operators we refer to [9]. □

The Trotter product formula omits any information on the rate of convergence of operator splitting methods. An approach in this direction has been made by Janke and Lubich in [10]. Before we state that theorem we need some knowledge about the error of quadrature formulae.

By our explanation in the appendix it suffices to analyse the error of a quadrature formula on the intervall  $[0, 1]$ .

**Definition 8.** *For  $x \in \mathbb{R}$  we denote  $x_+$  to be defined as  $x_+ = \begin{cases} x & \text{for } x \geq 0 \\ 0 & \text{for } x \leq 0 \end{cases}$*

**Definition 9.** *Let  $Q = (b_j, c_j)_{j=1}^m$  be a quadrature formula of order  $p$  and define the functions*

$$\varphi_\theta(t) = \frac{1}{(p-1)!} (t - \theta)_+^{p-1}$$

for  $\theta \in [0, 1]$ . Then the function

$$K_p(\theta) := E[\varphi_\theta]$$

is called the Peano kernel of  $Q$ .

**Theorem 3.** (Peano) Let  $Q[g]$  be a quadrature formula of order  $p \in \mathbb{N}$  and  $K_p : [0, 1] \rightarrow \mathbb{R}$  its Peano kernel. Then the error  $E[g]$  is given by

$$E[g] = \int_0^1 K_p(s)g^{(p)}(s)ds$$

for  $g \in C^p[0, 1]$ .

*Proof.* The Taylor expansion of  $g \in C^p[0, 1]$  gives

$$g(t) = h(t) + R_p(t)$$

where

$$h(t) = \sum_{\nu=0}^{p-1} \frac{g^{(\nu)}}{\nu!} \text{ and } R_p(t) = \int_0^t \frac{(t-\theta)^{p-1}}{(p-1)!} g^{(\nu)} d\theta.$$

Since  $E[g] = I[g] - Q[g]$ ,  $E : C^p[0, 1] \rightarrow \mathbb{R}$  is linear and therefore

$$E[g] = E[h + R_p] = E[h] + E[R_p] = E[R_p]$$

holds, because the quadrature formula integrates polynomials up to order  $p$  exactly. By Fubini it follows,

$$\begin{aligned} E[R_p] &= \int_0^1 \left( \int_0^t \frac{(t-\theta)^{p-1}}{(p-1)!} g^{(p)}(\tau) d\theta \right) dt - \sum_{j=1}^s b_j \int_0^{c_j} \frac{(c_j-\theta)^{p-1}}{(p-1)!} g^{(p)}(\theta) d\theta \\ &= \int_0^1 \left( \int_0^1 \frac{(t-\theta)_+^{p-1}}{(p-1)!} g^{(p)}(\theta) d\theta \right) dt - \sum_{j=1}^s b_j \int_0^1 \frac{(c_j-\theta)_+^{p-1}}{(p-1)!} g^{(p)}(\theta) d\theta \\ &= \int_0^1 \left( \int_0^1 \frac{(t-\theta)_+^{p-1}}{(p-1)!} g^{(p)}(\theta) dt \right) d\theta - \int_0^1 \sum_{j=1}^s b_j \frac{(c_j-\theta)_+^{p-1}}{(p-1)!} g^{(p)}(\theta) d\theta \\ &= \int_0^1 \left( \int_0^1 \frac{(t-\theta)_+^{p-1}}{(p-1)!} - \sum_{j=1}^s b_j \frac{(c_j-\theta)_+^{p-1}}{(p-1)!} \right) g^{(p)}(\theta) d\theta \\ &= \int_0^1 E[\varphi_\theta] g^{(p)}(\theta) d\theta \end{aligned}$$

□

**Theorem 4.** (Jahnke, Lubich)[10] Assume that there exists  $\alpha, \beta$  with  $0 \leq \alpha \leq 1 \leq \beta$  such that

$$\|[A, B]\psi\| \leq c_1 \|(-A)^\alpha \psi\| \quad (2.7)$$

and

$$\|[A, [A, B]]\psi\| \leq c_2 \|(-A)^\beta \psi\| \quad (2.8)$$

holds for all  $\psi$  in the domain of  $(-A)^\alpha$ . Then the local error of the Strang splitting is bounded by

$$\|e^{\frac{i\tau}{2}B} e^{i\tau A} e^{\frac{i\tau}{2}B} \psi - e^{i\tau(A+B)} \psi\| \leq C\tau^3 \|(-A)^\beta \psi\| \quad (2.9)$$

for all  $\psi$  in the domain of  $(-A)^\beta$ . The constant  $C$  depends only on  $c_1, c_2$  and  $\|B\|$ .

*Proof.* We begin by the duhamel formula

$$e^{-i\tau(A+B)} \psi = e^{-i\tau A} \psi - i \int_0^\tau e^{is_1 A} B e^{-i(\tau-s_1)(A+B)} \psi ds_1.$$

Reapplying the formula for the last exponential under the integral delivers

$$e^{-i\tau(A+B)} \psi = R_1 + R_2 + R_3$$

with

$$R_1 = e^{-i\tau A} \psi - i \int_0^\tau e^{is_1 A} B e^{-i(\tau-s_1)A} \psi ds_1$$

$$R_2 = (-i)^2 \int_0^\tau e^{-is_1 A} B \int_0^{\tau-s_1} e^{-is_2 A} B e^{-i(\tau-s_1-s_2)A} \psi ds_2 ds_1 \text{ and}$$

$$R_3 = (-i)^3 \int_0^\tau e^{-is_1 A} B \int_0^{\tau-s_1} e^{-is_2 A} B \int_0^{\tau-s_1-s_2} e^{-is_3 A} B e^{-i(\tau-s_1-s_2-s_3)(A+B)} \psi ds_3 ds_2 ds_1$$

where  $R_3$  is bounded by  $\|R_3\| \leq C_r \tau^3 \|B\| \|\psi\| = C_r \tau^3 \|B\|$ . since  $\|\psi\| = 1$

On the other hand the exponential series of  $e^{-i\frac{\tau}{2}B}$  gives

$$e^{\frac{i\tau}{2}B} e^{i\tau A} e^{\frac{i\tau}{2}B} \psi = S_1 + S_2 + S_3$$

with

$$S_1 = e^{-i\tau A} \psi - \frac{i\tau}{2} (e^{-i\tau A} B + B e^{-i\tau A}) \psi$$

$$S_2 = \frac{-i\tau}{8} (e^{-i\tau A} B^2 + 2B e^{-i\tau A} B + B^2 e^{-i\tau A}) \psi$$

$$S_3 = \frac{(-i\tau)^3}{48} (4B e^{-i\tau A} B^2 + 4B^2 e^{-i\tau A} B + e^{-i\tau A} B^3) \psi + \mathcal{O}(\tau^4)$$

where  $S_3$  is bounded by  $\|S_3\| \leq C_s \tau^3 \|B\|^3$ .

Putting all together, we have

$$e^{-i\tau(A+B)}\psi - e^{-\frac{i\tau}{2}B}e^{-i\tau A}e^{-\frac{i\tau}{2}B}\psi = S_1 - R_1 + S_2 - R_2 + S_3 - R_3$$

where  $S_3 - R_3$  is bounded by  $\|S_3 - R_3\| \leq C\tau^3\|B\|^3$ .

It remains to study

$$r_1 = S_1 - R_1$$

and

$$r_2 = S_2 - R_2$$

Define the function

$$f(s_1) := e^{-is_1A}Be^{-i(\tau-s_1)A}\psi.$$

Then

$$r_1 = -i \left[ \frac{\tau}{2} (f(0) + f(\tau)) - \int_0^\tau f(s_1) ds_1 \right].$$

The term inside the squared bracket is the error of the trapezoidal rule. With the Peano kernel-theorem we can write

$$r_1 = \frac{i\tau^3}{2} \int_0^1 \theta(1-\theta) f''(\theta\tau) d\theta$$

where  $K_2(\theta) = -\frac{1}{2}\theta(1-\theta)$  is the second order Peano kernel of the trapezoidal rule.

$$\begin{aligned} f''(s_1) &= (i)^2 A^2 e^{-is_1A} B e^{-i(\tau-s_1)A} \psi - i^2 e^{-is_1A} B e^{-i(\tau-s_1)A} A \psi \\ &\quad - i^2 A e^{is_1A} B e^{-i(\tau-s_1)A} A \psi + i^2 e^{-is_1A} B e^{-i(\tau-s_1)A} A^2 \psi \\ &= e^{-is_1A} (-A^2 B + 2ABA - BA^2) e^{-i(\tau-s_1)A} \psi \text{ since } [e^{-itA}, A] = 0 \\ &= e^{-is_1A} [[A, B], A] e^{-i(\tau-s_1)A} \psi \\ &= -e^{-is_1A} [A, [A, B]] e^{-i(\tau-s_1)A} \psi \end{aligned}$$

Combining  $f''(\theta\tau) = -e^{-i\theta\tau A} [A, [A, B]] e^{-i(\tau-\theta\tau)A} \psi$  and condition (2.8) we have

$$\|r_1\| \leq \frac{\tau}{2} \int_0^1 \theta(1-\theta) \|f''(\theta\tau)\psi\| d\theta = \frac{1}{12} c_2 \tau^3 \|(-A)^\beta \psi\|$$

Thus it remains to study  $r_2$ . For that purpose define the function

$$g(s_1, s_2) = e^{-is_1A} B e^{-is_2A} B e^{-i(\tau-s_1-s_2)A} \psi.$$

Then we can write

$$r_2 = i^2 \left( \frac{(\tau)^2}{8} (g(0,0) + 2g(0,\tau) + g(\tau,0)) - \int_0^\tau \int_0^{\tau-s_1} g(s_1, s_2) ds_2 ds_1 \right),$$

which is the error of a quadrature formula that integrates constant functions exactly. Therefore  $r_2$  is bounded by

$$\|r_2\| \leq \tilde{c}\tau^3 \left( \max\left\|\frac{\partial g}{\partial s_1}\right\| + \max\left\|\frac{\partial g}{\partial s_2}\right\| \right)$$

where the maxima are taken over the triangle  $0 \leq s_1 \leq \tau$ ,  $0 \leq s_2 \leq \tau - s_1$ .

$$\begin{aligned} \frac{\partial g}{\partial s_1}(s_1, s_2) &= -ie^{-is_1A}ABe^{-is_2A}Be^{-i(\tau-s_1-s_2)A}\psi + ie^{-is_1A}Be^{-is_2A}BAe^{-i(\tau-s_1-s_2)A}\psi \\ &= ie^{-is_1A}(-ABe^{-is_2A}B + Be^{-is_2A}BA)e^{-i(\tau-s_1-s_2)A}\psi \\ &= -ie^{-is_1A}([A, B]e^{-is_2A}B + Be^{-is_2A}[A, B])e^{-i(\tau-s_1-s_2)A}\psi. \end{aligned}$$

Since  $[A, B] = -[B, A]$  and

$$[B, A]e^{-is_2A}B + Be^{-is_2A}[B, A] - (-ABe^{-is_2A}B + Be^{-is_2A}BA) = 0.$$

Now we can bound

$$\frac{\partial g}{\partial s_1}(s_1, s_2) = -i(e^{-is_1A}[A, B]e^{-is_2A}Be^{-i(\tau-s_1-s_2)A} + e^{-is_1A}Be^{-is_2A}[A, B]e^{-i(\tau-s_1-s_2)A})\psi$$

using (2.7) with  $\alpha = 1$  by

$$\left\|\frac{\partial g}{\partial s_1}\right\| \leq 2c_1\|A\psi\|\|B\psi\|.$$

Similarly,

$$\frac{\partial g}{\partial s_2}(s_1, s_2) = ie^{-is_1A}Be^{-is_2A}[A, B]e^{-i(\tau-s_1-s_2)A}\psi$$

is bounded by

$$\left\|\frac{\partial g}{\partial s_2}\right\| \leq c_1\|A\psi\|\|B\psi\|.$$

By substitution of the norm of  $B$  inside the constant  $C_2$  we obtain

$$\|r_2\| \leq C_2\tau^3\|A\psi\|.$$

Summation of the parts gives

$$\begin{aligned} \|e^{\frac{i\tau}{2}B}e^{i\tau A}e^{\frac{i\tau}{2}B}\psi - e^{i\tau(A+B)}\psi\| &\leq \frac{1}{12}c_2\tau^3\|(-A)^\beta\psi\| + C_2\tau^3\|A\psi\| + C_1\tau^3\|B\|^3 \\ &\leq C\tau^3\|(-A)^\beta\psi\| \end{aligned}$$

which completes the proof. □

## 2.3 Stability and energy conservation

Since  $\Phi(\tau)$  is an unitary operator we have

$$\|\Phi(\tau)^n \psi\| \leq \|\Phi(\tau)\|^n \|\psi\| \leq 1$$

for all  $\psi$  with  $\|\psi\| = 1$  and  $k = 1, \dots, T/\tau$  as  $\tau \rightarrow 0, h \rightarrow 0$ , which yields that the symmetric operator splitting is unconditionally stable. As an unitary operator the symmetric operator splitting method has also the property to conserve the  $l^2$ -norm of the approximated solutions.

It is an efficient and accurate method preferable if  $[H_0, V] = 0$ . For in that case  $\Phi(\tau)$  it is energy conserving, since if  $[H_0, V] = 0$  then  $[e^{-itH_0}, V] = 0$  and  $[e^{-itV}, H_0] = 0$ . It follows,

$$(\psi^{k+1}, H\psi^{k+1}) = (\psi^k, \Phi^*(\tau)H\Phi(\tau)\psi^k) = (\psi^k, H\psi^k).$$

The next Theorem states that the symmetric operator splitting is not energy conserving if  $[H_0, V] \neq 0$ .

**Theorem 5.** *The symmetric operator splitting does not conserve energy, more precisely  $(\psi^{k+1}, H\psi^{k+1}) = (\psi^k, H\psi^k) + \mathcal{O}(\tau^2)$ , for all  $k \in \mathbb{N}$  with  $k\tau \leq T$ .*

*Proof.*

$$\begin{aligned} \Phi^* &= e^{\frac{i\tau}{2}V} e^{-i\tau H_0} e^{\frac{i\tau}{2}V} = \left(1 + \frac{i\tau}{2}V\right) (1 - i\tau H_0) \left(1 + \frac{i\tau}{2}V\right) + \mathcal{O}(\tau^2) \\ &= \left(1 + \frac{i\tau}{2}V - i\tau H_0 + \frac{i\tau}{2}V\right) + \mathcal{O}(\tau^2). \end{aligned}$$

Similarly,

$$\begin{aligned} \Phi &= e^{-\frac{i\tau}{2}V} e^{i\tau H_0} e^{-\frac{i\tau}{2}V} = \left(1 - \frac{i\tau}{2}V\right) (1 + i\tau H_0) \left(1 - \frac{i\tau}{2}V\right) + \mathcal{O}(\tau^2) \\ &= \left(1 - \frac{i\tau}{2}V + i\tau H_0 - \frac{i\tau}{2}V\right) + \mathcal{O}(\tau^2). \end{aligned}$$

It follows,

$$\begin{aligned} \Phi^* H \Phi &= \left(1 + \frac{i\tau}{2} - i\tau H_0 + \frac{i\tau}{2}V\right) \left(H - \frac{i\tau}{2}HV + i\tau H H_0 - \frac{i\tau}{2}HV\right) + \mathcal{O}(\tau^2) \\ &= \left(H - \frac{i\tau}{2}HV + i\tau H H_0 - \frac{i\tau}{2}HV + \frac{i\tau}{2}VH - i\tau H_0 H + \frac{i\tau}{2}VH\right) + \mathcal{O}(\tau^2) \\ &= H - i\tau (HV - VH - H H_0 + H_0 H) + \mathcal{O}(\tau^2) \\ &= H - i\tau (H(-H_0 + V) - (-H_0 + V)H) + \mathcal{O}(\tau^2) = H + \mathcal{O}(\tau^2). \end{aligned}$$

And therefore,

$$\begin{aligned}
 (\psi^{k+1}, H\psi^{k+1}) &= (\Phi\psi^k, H\Phi\psi^k) = (\psi^k, \Phi^*H\Phi\psi^k) = (\psi^k, H\psi^k + \mathcal{O}(\tau^2)) \\
 &= (\psi^k, H\psi^k) + \mathcal{O}(\tau^2).
 \end{aligned}$$

□

Therefore, the symmetric operator splitting of order two conserves energy up to order one with respect to the temporal discretisation parameter  $\tau$ .

High-order splitting schemes of order  $p$  was constructed by Suzuki in [12] and Yoshida in [13].

$$e^{\tau(A+B)} = e^{a_1\tau A} e^{b_1\tau B} e^{a_2\tau A} e^{b_2\tau B} \dots e^{a_s\tau A} e^{b_s\tau B} + \mathcal{O}(\tau^r)$$

where  $r$  depends on both  $a_1, \dots, a_s$  and  $b_1, \dots, b_s$  which can be chosen real or complex valued.

A Trotter-Suzuki splitting method of order  $p$  with  $l$  coefficients needs  $2l$  Fourier transforms and  $l$  vector by vector multiplications for one time step propagation. One may argue that for large  $l$  the method is therefore as fast as methods that compute the solution at one time step by a matrix-vector product. But still, high-order splitting schemes are preferable, because the components of those schemes are not only cheap to compute but also require less storage memory, which makes operator splitting methods quite attractive for long time evolution problems.

### 3 The implicit Crank-Nicholson Method

Both the explicit Crank-Nicholson method and the implicit one are schemes of order two of space and time. However, the implicit CN method has the advantage that it is unconditionally stable, one is free in the choice of the mesh width in terms of stability.

#### 3.1 Construction

There are different ways to obtain the propagation matrix  $\Phi$  of the Crank-Nicholson scheme. The way chosen here gives a better insight and reflects the original idea. Direct discretisation of the differential operators and averaging between two adjacent points in time yields

$$i \frac{\psi_j^{k+1} - \psi_j^k}{\tau} = \frac{1}{2} \left[ -\frac{1}{2} \frac{\psi_{j+1}^{k+1} - 2\psi_j^{k+1} + \psi_{j-1}^{k+1}}{h^2} + V_j \psi_j^{k+1} - \frac{1}{2} \frac{\psi_{j+1}^k - 2\psi_j^k + \psi_{j-1}^k}{h^2} + V_j \psi_j^k \right].$$

In matrix form it may be written as

$$\psi^{k+1} = \psi^k - \frac{i\tau}{2} H(\psi^k + \psi^{k+1})$$

with the discretized Hamiltonian operator

$$H = -\frac{1}{2} \frac{\delta_{c+1,d} - 2\delta_{c,d} + \delta_{c-1,d}}{h^2} + V_c \delta_{c,d},$$

where

$$\delta_{c,d} = \begin{cases} 1 & \text{if } c = d \\ 0 & \text{elsewise.} \end{cases}$$

Therefore we have

$$\left( I + \frac{i\tau}{2} H \right) \psi^{k+1} = \left( I - \frac{i\tau}{2} H \right) \psi^k$$

and finally with

$$\Phi(\tau) = \left( I + \frac{i\tau}{2} H \right)^{-1} \left( I - \frac{i\tau}{2} H \right)$$

it follows

$$\psi^{k+1} = \Phi(\tau) \psi^k.$$

It is advisable to employ the band structure of the Hamiltonian and use sparse matrix techniques to compute the propagation matrix  $\Phi(\tau)$ . Since  $H$  is tridiagonal we can improve the performance of the Crank-Nicholson method by avoiding to compute the matrix inverse. Therefore let us rewrite  $\Phi(\tau)$  as

$$\begin{aligned}\psi^{k+1} &= \left(I + \frac{i\tau}{2}H\right)^{-1} \left[2I - \left(I + \frac{i\tau}{2}H\right)\right] \psi^k \\ &= \left[2\left(I + \frac{i\tau}{2}H\right)^{-1} - I\right] \psi^k\end{aligned}$$

Now we can avoid to compute the matrix inversion by splitting our problem in two steps. First we solve the linear system

$$\left(I + \frac{i\tau}{2}H\right) \chi = \frac{1}{2}\psi^k$$

then we update the solution as

$$\psi^{k+1} = \chi - \psi^k.$$

Note that by avoiding to compute a matrix inverse, we have to solve a system of linear equations at every time step.

**Lemma 6.** *The implicit Crank-Nicholson method is consistent with the differential equation (1.1).*

*Proof.* Let  $\phi^k$  be the exact solution of the Schrödinger equation at time  $t = t_0 + k\tau$ . Then the local error is given by

$$\begin{aligned}\|\phi^{k+1} - \frac{2I - i\tau H}{2I + i\tau H}\phi^k\| &= \left\| \left( e^{-i\tau H} - \frac{2I - i\tau H}{2I + i\tau H} \right) \phi^k \right\| = \\ &= \left\| \left( (2I + i\tau H) e^{-i\tau H} - (2I - i\tau H) \right) \phi^k \right\| = \mathcal{O}(\tau^3)\end{aligned}$$

as the first term inside the norm can be written as

$$\begin{aligned}(2I + i\tau H) e^{-i\tau H} &= (2I + i\tau H) \left( I - i\tau H + \frac{(i\tau)^2}{2}H^2 + \mathcal{O}(\tau^3) \right) = \\ &= 2I - i\tau H + \mathcal{O}(\tau^3)\end{aligned}$$

and  $\|\phi^k\| = 1$  for all  $k$ . Therefore the Crank-Nicholson method is consistent of order two with the Schrödinger equation.  $\square$

## 3.2 Stability

Denote the spectral radius of a matrix  $A$  by  $\rho(A)$ . Since the discrete Hamiltonian is real and symmetric it follows

$$\|H\| = \sqrt{\rho(H^T H)} = \sqrt{\rho(H^2)} = \sqrt{\rho^2(H)} = \rho(H) \quad (3.1)$$

**Lemma 7.** *The Crank-Nicholson method applied to the equation (1.1) is unconditionally stable.*

*Proof.* Let  $\Phi = (2I - i\tau H)^{-1}(2I + i\tau H)$  represent the short time propagations matrix of the CN-scheme. Since  $\Phi$  is an unitary matrix

$$\Phi^* \Phi = \frac{2I - i\tau H}{2I + i\tau H} \frac{2I + i\tau H}{2I - i\tau H} = 1$$

it follows

$$\|\Phi^k \psi\| \leq \|\Phi\|^k \|\psi\| \leq 1$$

for all  $\psi$  with  $\|\psi\| = 1$  and  $k = 1, \dots, T/\tau$  as  $\tau \rightarrow 0, h \rightarrow 0$ . □

Unconditionally stable methods have the property that their stability does not depend on size of the discretisation parameter  $\tau$ . However, because of the consistency error one is not completely free in the choice of  $\tau$  if accurate solutions are preferred.

We have shown that the Crank-Nicholson method is consistent with the Schrödinger equation, stable and therefore, by Lax-equivalence theorem, convergent. The next theorem shows that the Crank-Nicholson method has the desired property of energy conservation.

**Theorem 8.** *The CN-scheme conserves energy.*

*Proof.*

$$\begin{aligned} (\psi^{n+1}, H\psi^{n+1}) &= (\Phi\psi^n, H\Phi\psi^n) = (\psi^n, \Phi^* H\Phi\psi^n) = \\ &= (\psi^n, \frac{2I + i\tau H}{2I - i\tau H} H \frac{2I - i\tau H}{2I + i\tau H} \psi^n) = (\psi^n, H \frac{2I + i\tau H}{2I - i\tau H} \frac{2I - i\tau H}{2I + i\tau H} \psi^n) = \\ &= (\psi^n, H\psi^n) \end{aligned}$$

□

The magnitude of energy conservation is  $\mathcal{O}(\tau^2)$  for the Crank-Nicholson method. The next method we want to introduce has a higher order of energy conservation. Basically one is able to construct methods of arbitrary order of accuracy and energy conservation.

## 4 Method of Scaling and Squaring

The Crank-Nicholson approximation of the matrix exponential shows a low order of consistency. To overcome this problem we introduce a general concept of approximating (matrix valued) functions known as the Padé approximation.

**Definition 10.** For any function  $g(z)$  analytic in a neighborhood of  $z = 0$ , one defines its Padé approximants  $r[k, m](z)$  as the rational function

$$r[k, m](z) = \frac{p(z)}{q(z)}, \quad p \in \mathbb{P}_m, \quad q \in \mathbb{P}_n,$$

satisfying

$$g(z)q(z) - p(z) = \mathcal{O}(z^{n+m+1}) \quad \text{as } z \rightarrow 0.$$

The rational function  $r[k, m]$  is uniquely determined by this definition, although in some cases,  $p$  and  $q$  may have common factors. If  $p$  and  $q$  are irreducible one can assume without loss of generality that  $q(0) = 1$ . [3]

The benefit of the scaling and squaring method is the fact that the  $[k, m]$  Padé approximants to the exponential function are known explicitly for all  $k$  and  $m$ .

**Theorem 9.** The Padé approximant  $r_{km}$  to the exponential function  $g(z) = e^z$  is given by

$$r_{km}(z) = \frac{p_{km}(z)}{q_{km}(z)},$$

where

$$p_{km}(z) = \sum_{j=0}^k \frac{(k+m-j)!k!}{(k+m)!j!(k-j)!} (z)^j$$

and

$$q_{km}(z) = \sum_{j=0}^m \frac{(k+m-j)!m!}{(k+m)!j!(m-j)!} (-z)^j,$$

moreover

$$e^z - \frac{p[k, m](z)}{q[k, m](z)} = C_{k,m} z^{k+m+1} + \dots,$$

where

$$C_{k,m} = (-1)^k \frac{k!m!}{(k+m)!(n+m+1)!}.$$

The proof can be found in ([4], theorem 5.5.1).

Now we can approximate the exponential of a matrix by substitution of a matrix  $A \in \mathbb{C}$  for the variable  $z$  in the Padé approximants. This approximation is reliable near the origin, that is, for small  $\|A\|$ . Therefore, in order to reduce the norm of  $A$  one exploits the relation  $e^A = (e^{A/\sigma})^\sigma$  for a  $\sigma \in \mathbb{C}$ . The idea is to choose  $\sigma = 2^s$  such that  $\|A/\sigma\| < 1$ , approximate  $e^{A/2^s} \approx r_{km}(A/2^s)$  and then undo the scaling by  $s$  repeated squarings, that is to take  $e^A \approx r_{km}(A/2^s)^{2^s}$ . Diagonal Padé approximants, that is  $k = m$ , are preferred because  $r_{km}$  is less accurate than  $r_{jj}$  when  $j = \max(k, m)$  at the same computational cost. In addition, if the eigenvalues of the matrix  $A$  lie in the open left half-plane then for the spectral radius we have  $\rho(r_m(A)) < 1$ , where  $r_m(A)$  denotes the diagonal Padé approximant of order  $m$  to the matrix exponential. This is an important property in applications to stiff differential equations.

An investigation of the effects of approximation errors in the Padé approximant has been done by many authors, recently by N. Higham in [11]. In that paper Higham gives a new backward error analysis in exact arithmetic and deduces an implementation of optimal efficiency.

A backward error analysis is given by the following theorem. It states that the truncation errors in the Padé approximant are equivalent to a perturbation of the original matrix.

**Theorem 10.** [11] *Let the diagonal Padé approximant  $r_m$  satisfy*

$$e^{-2^{-s}A} r_m(2^{-s}A) = I + G, \quad (4.1)$$

where  $\|G\| < 1$ . Then

$$r_m(2^{-s}A)^{2^s} = e^{A+E},$$

where  $E$  commutes with  $A$  and

$$\frac{\|E\|}{\|A\|} \leq \frac{-\log(1 - \|G\|)}{\|2^{-s}A\|}.$$

*Proof.* Let

$$e^{-A} r_m(A) = I + G = e^H,$$

where we assume that  $\|G\| < 1$ , so that  $\log(I + G)$  exists. From the series representation  $\log(I + G) = \sum_{j=1}^{\infty} (-1)^{j+1} G^j/j$ , we get

$$\|H\| = \|\log(I + G)\| \leq \sum_{j=1}^{\infty} \|G\|^j/j = -\log(1 - \|G\|).$$

Since  $G$  is a function of  $A$ , in the sense of matrix functions, so is  $H$ . Therefore  $H$  commutes with  $A$ . It follows that

$$r_m(A) = e^A e^H = e^{A+H}.$$

Now we replace  $A$  by  $A/2^s$ , with a nonnegative integer  $s$  and raise both sides of this equation to the power of  $2^s$  to obtain

$$r_m (A/2^s)^{2^s} = e^{A+E},$$

where  $E = 2^s H$  satisfies

$$\|E\| \leq -2^s \log(1 - \|G\|).$$

A division on both sides of the equation by  $\|2^s A\|$  leads to the claim. □

The elegance of backward error analysis is that we do not have to consider the conditioning of the problem. The error that arises by using the Padé approximants plus the error arising from the scaling and squaring phase are automatically taken into account and therefore we do not have to consider the conditioning of the matrix exponential problem. However, a thorough discussion on the condition of the matrix exponential problem can be found in [3] and [4].

In order to see that the method of scaling and squaring is stable, observe that  $\Phi(\tau) = r_{mm}(-itH)$  is a unitary matrix.

$$\Phi^*(\tau)\Phi(\tau) = \frac{\sum_{j=1}^p c_j(-i\tau H)^j}{\sum_{j=1}^p c_j(i\tau H)^j} \frac{\sum_{j=1}^p c_j(i\tau H)^j}{\sum_{j=1}^p c_j(-i\tau H)^j} = I$$

and therefore,

$$\|\Phi^n\| \leq \|\Phi\|^n \leq 1$$

is valid for all  $\psi$  with  $\|\psi\| = 1$  and  $k = 1, \dots, T/\tau$  as  $\tau \rightarrow 0, h \rightarrow 0$ . Thus the method of scaling and squaring using diagonal Padé approximants is unconditionally stable. Note that the method of scaling and squaring is by definition consistent with the Schrödinger equation,

$$\|(e^{-itH} - r_{mm}(e^{-itH}))\psi\| \leq \|\psi\| + \mathcal{O}(\tau^{2m}),$$

and therefore by Lax equivalence-theorem it follows that the method of scaling and squaring using diagonal Padé approximants is convergent.

**Theorem 11.** *The Scaling and Squaring method, using diagonal Padé-approximants, conserves energy.*

*Proof.* As a first step we have

$$\frac{\sum_{j=1}^p c_j(i\tau H)^j}{\sum_{j=1}^p c_j(-i\tau H)^j} = \frac{\sum_{j=1}^p (i\tau)^j c_j H^j}{\sum_{j=1}^p (-i\tau)^j c_j H^j} = \frac{\sum_{j=1}^p (i\tau)^j c_j H^j}{(\sum_{j=1}^p (i\tau)^j c_j H^j)^*}.$$

The  $c_j$  are real coefficients from the diagonal Padé approximant for the Matrix exponential. Therefore,

$$\begin{aligned}
 (\psi^{n+1}, H\psi^{n+1}) &= \left( \frac{\sum_{j=1}^p (i\tau)^j c_j H^j}{(\sum_{j=1}^p (i\tau)^j c_j H^j)^*} \psi^n, H \frac{\sum_{j=1}^p (i\tau)^j c_j H^j}{(\sum_{j=1}^p (i\tau)^j c_j H^j)^*} \psi^n \right) \\
 &= \left( \psi^n, \frac{(\sum_{j=1}^p (i\tau)^j c_j H^j)^*}{(\sum_{j=1}^p (i\tau)^j c_j H^j)} H \frac{\sum_{j=1}^p (i\tau)^j c_j H^j}{(\sum_{j=1}^p (i\tau)^j c_j H^j)^*} \psi^n \right) \\
 &= \left( \psi^n, H \frac{(\sum_{j=1}^p (i\tau)^j c_j H^j)^*}{(\sum_{j=1}^p (i\tau)^j c_j H^j)} \frac{\sum_{j=1}^p (i\tau)^j c_j H^j}{(\sum_{j=1}^p (i\tau)^j c_j H^j)^*} \psi^n \right) = (\psi^n, H\psi^n).
 \end{aligned}$$

□

**Remark 1.** *The Crank-Nicholson method can be derived from the (1,1) Padé approximation to the matrix exponential.*

## 5 Testing

In this chapter we test the methods discussed above. For that we introduce various test cases of different type, that exhibit the power and weaknesses of our numerical integrators respectively. The three integrators  $\Phi_{CN}$ ,  $\Phi_{Spl}$ ,  $\Phi_{pad}$  coming from the Crank-Nicholson method, the symmetric operator splitting method and the method of scaling and squaring respectively, will be tested for different potentials  $V(x)$  in terms of consistency, energy conservation and reliability.

### 5.1 A free particle

By applying the foregoing numerical integrators in the case of a free particle, i.e.  $V(x) = 0 \quad \forall x \in X = [-L, L]$ , we gain a first insight of their qualitative behavior. In that case, the solution of the Schrödinger equation is formally given by

$$\psi(x, t) = \exp(-i(t - t_0)H_0) \psi(x, t_0).$$

We may assume  $t_0 = 0$ . The initial wave function is a normalized Gaussian wave packet

$$\psi^0 = \psi(x, 0) = \frac{1}{\sqrt{\sigma_0} \sqrt{\pi}} \exp(ik_0 x) \exp\left(\frac{-(x - x_0)^2}{2\sigma_0^2}\right)$$

localized about  $x_0$  with a packet width of  $\sigma_0$  and an average momentum  $p_0 = \hbar k_0$ . In free space the Gaussian evolves as

$$\psi(x, t) = \frac{1}{\sqrt{\sigma_0} \sqrt{\pi}} \frac{\sigma_0}{\alpha} \exp\left(ik_0 \left(x - \frac{p_0 t}{m}\right)\right) \exp\left(-\frac{(x - x_0 - p_0 t/m)^2}{2\sigma_0^2 + i\hbar t/m}\right)$$

with  $\alpha^2 = \sigma_0^2 + i\hbar t/m$ . The probability density  $P(x, t) = |\psi(x, t)|^2$  is

$$P(x, t) = \frac{\sigma_0}{|\alpha|^2 \sqrt{\pi}} \exp\left(-\left(\frac{\sigma_0}{|\alpha|}\right)^4 \frac{(x - x_0 - p_0 t/m)^2}{\sigma_0^2}\right).$$

By symmetry, the maximum of the Gaussian equals the expectation value  $\langle x \rangle = \int_{-\infty}^{\infty} x P(x, t) dx$ . It moves in time as  $\langle x \rangle = x_0 + p_0 t/m$ , that is, the packet moves with a velocity  $p_0/m$ , where  $m$  denotes the mass of the particle here. Due to dispersion the Gaussian spreads in time with a standard deviation

$$\sigma(t) = \sigma_0 \sqrt{\left(\frac{|\alpha|}{\sigma_0}\right)^4} = \sigma_0 \sqrt{1 + \frac{\hbar^2 t^2}{m^2 \sigma_0^4}}.$$

Since the Gaussian is normalized, the energy is given by

$$(\psi, H_0\psi) = (\psi^0, H_0\psi^0) = \frac{\hbar^2 k_0^2}{2m}.$$

For  $V(x) = 0 \forall x \in X$  the methods discussed above are energy conserving.

To be able to compare the methods above we may fix some quantities for the testing phase. Let  $L = 250$ ,  $\hbar = 1$  and  $m = 1$ . The space  $X = [-L, L]$  is discretised by  $m = X/h$  equidistant gridpoints, where the meshwidth  $h$  is given by  $h = \pi/50$ . The choice of the meshwidth depends on two factors. The smaller one chooses  $h$  the bigger the size of the Hamiltonian  $H$ , and therefore the computation of the matrix exponential needs more computer time and storage memory. On the other hand, if  $h$  is chosen large, the approximated solutions would differ seriously from the exact ones. Our initial wave packet is the Gaussian:

$$\psi(x, 0) = \frac{1}{\sqrt{10\sqrt{\pi}}} \exp(ix) \exp\left(-\frac{(x - 150)^2}{200}\right)$$

with wave number  $k_0 = 1$ , localized at  $x_0 = -150$  and the deviation  $\sigma_0 = 10$ .

The boundary conditions are periodic, more precisely  $\psi(0, t) = \psi(X, t) = 0$ . Note that with these boundary conditions we introduce infinitely high potential walls at the boundary. However, as we want the support of the propability density function  $P(x, t)$  to be convex in  $X$  for all times  $t$ , we choose  $T$  as  $T = \max\{t \mid \text{support of } P(x, t) \text{ is convex in } X\}$ . That is, we want the Gaussian to stay in  $X$  as long as possible without hitting the boundary.

The Crank-Nicholson method  $\Phi_{CN}$  is applied in the usual way. The performance can be improved, if we utilise that  $H$  is sparse. Then, we need to solve a linear system at every time step.

We use Matlab's built-in routine *expm* in order to test the method of scaling and squaring  $\Phi_{pad}$ . It uses a  $R_{66}$  Padé approximation of the matrix exponential and  $\|2^{-s}A\|_{\infty} \leq 0.5$  as the scaling criterion. Moreover it requires five matrix multiplication because it evaluates all the powers  $A^2, A^3, \dots, A^6$ .

The accuracy of operator splitting methods  $\Phi_{Spl}$  in the case of a free particle is higher. In momentum space the solution of the Schrödinger equation for a free particle evolves as

$$\hat{\psi}(k, t) = \exp\left(-it\frac{k^2}{2}\right) \hat{\psi}(k, 0)$$

which can be computed very efficiently at any time  $t$ . Then Fourier transform is applied in order to evaluate the solution in position space.

The momentum space must be discretized according to the Nyquist-Shannon Sampling theorem to avoid aliasing. The critical frequency is given by  $K_c = \pi/h$ . Then for a particle we have to determine the wave number  $k = p/\hbar \in [-K_c, K_c]$ .

The solution of the free Schrödinger equation computed by each of the methods  $\Phi_{pad}$ ,  $\Phi_{CN}$ ,  $\Phi_{Spl}$ , is compared with the exact solution at adequate time points. More precisely let  $\psi(x_j, t_k)$  be the solution of the Schrödinger equation computed by  $\Phi_{(\cdot)}$  and  $\phi(x_j, t_k)$  represent the exact solution at  $(x_j, t_k)$ . Then the function

$$d(t) = \|\psi(x_j, t_k) - \phi(x_j, t_k)\|_h$$

measures the consistency of the method  $\Phi_{\cdot}$ . Additionally we define the energy of the particle at time  $t_k$  as

$$E(\tau) = h \sum_{j=1}^m (\psi^k (H\psi^k)^*)_j.$$

In the following figures the legend 'expm' belongs to the method of scaling and squaring  $\Phi_{pad}$ .

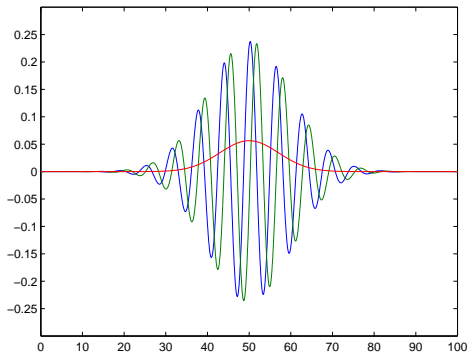


Figure 5.1: The initial condition  $\psi(x, 0)$

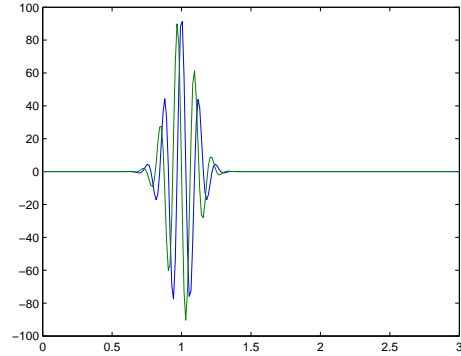


Figure 5.2: Fourier transform of  $\psi(x, 0)$

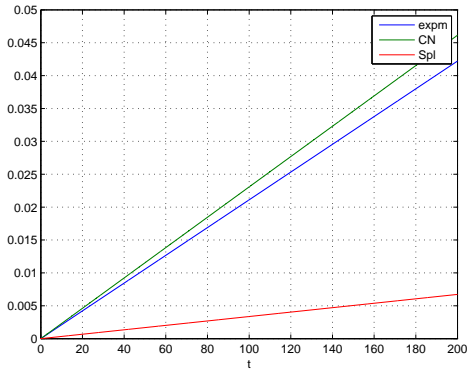


Figure 5.3: Error  $d(t)$  for  $\Phi_{CN}$ ,  $\Phi_{Spl}$  and  $\Phi_{Pad}$

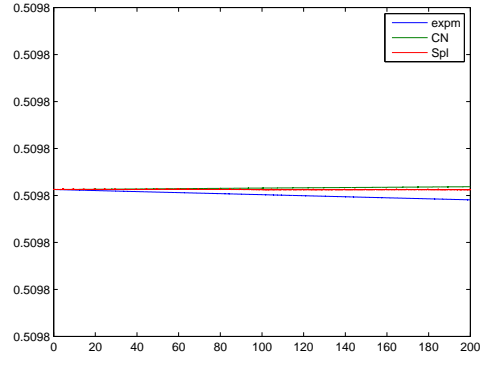


Figure 5.4: Energy  $E(\tau)$  vs. time for  $\Phi_{CN}$ ,  $\Phi_{pad}$  and  $\Phi_{Spl}$

Figure 5.3 shows the error function  $d(t)$  with respect to the methods discussed above. We see that the Crank-Nicholson scheme and the method of scaling and squaring

behave very alike. We can also observe that both the Crank-Nicholson scheme and the scaling and squaring method are less accurate than the symmetric operator splitting method. That is, for  $V = 0$ , the solution of the Schrödinger equation can be computed in an easy and cheap way.

Tests have shown, that we can enlarge  $L$  up to  $L = 300$ . Therefore, we can choose the total physical time  $T = 350$ ; The mesh width in space  $h = \pi/50$  has been retained unchanged to guarantee  $\max_{t \leq T} d(t) \leq 0.1$  for all  $t \in [0, T]$ . In that case the Hamiltonian is a matrix of the size 9550 approximately, which means that we need about 729.62 Mbyte memory on a 64 bit machine in order to store it.

The other methods allow  $h$  to be chosen smaller, particularly in the case of the symmetric exponential splitting method we are not explicitly depending on the choice of  $h$ . The meshwidth in time remains fixed as  $\tau = 1/30$ .

As expected, the solutions delivered by the scaling and squaring method as well as the Crank-Nicholson method are more accurate if  $h$  is chosen small. Figure 5.3 shows the consistency error function  $d(t)$  for the methods under consideration. Again, both the method of scaling and squaring and the Crank-Nicholson method show similar behavior while the symmetric exponential operator splitting method is the most accurate method for the free space scenario.

## 5.2 Coherent states of the harmonic Oscillator

Coherent states of the harmonic oscillator are specific quantum states. The dynamics of these quantum states resemble closely to those of the classical harmonic oscillator. In order to introduce these particular states we consider the time-dependent Schrödinger equation for a particle with unit mass in a quadratic potential. In other words, the Hamiltonian has the form

$$H = -\frac{\hbar}{2} \frac{d^2}{dx^2} + \frac{\omega^2}{2} x^2.$$

For the stationary states the mean value of the particles position vanishes, i.e.  $\langle x \rangle = 0$ . Quantum states of the harmonic oscillator for which  $\langle x \rangle \neq 0$  are eigenfunctions of the annihilation operator  $a$ , i.e.

$$a\phi_\alpha = \alpha\phi_\alpha$$

with a complex number  $\alpha \neq 0$ .

If we write  $\alpha = |\alpha|e^{i\delta}$ ,  $|\alpha| = 5$ ,  $\delta = \frac{\pi}{2}$ ,  $\sigma = \sqrt{\frac{\hbar}{\omega}}$  and  $\omega = \frac{2\pi}{900}$  with the initial wave function represented by a gaussian

$$\phi(x, 0) = \frac{1}{\sqrt[4]{\pi}\sqrt{\sigma}} \exp\left(i\frac{\sqrt{2}5x}{\sigma}\right) \exp\left(-\frac{1}{2\sigma^2}x^2\right).$$

then

$$\phi(x, t) = \frac{1}{\sqrt[4]{\pi}\sqrt{\sigma}} \exp\left(-i\frac{\omega t}{2}\right) \exp\left(i\left[\frac{|\alpha|^2}{2} \sin 2(\omega t - \delta) - \frac{\sqrt{2}|\alpha|x}{\sigma} \sin(\omega t - \delta)\right]\right) \exp\left(-\frac{1}{2\sigma^2} \left(x - \sigma\sqrt{2}|\alpha| \cos(\omega t - \delta)\right)^2\right)$$

are the coherent states of the quantum harmonic oscillator. For the probability density we get

$$P(x, t) = |\phi(x, t)|^2 = \frac{1}{\sqrt{\pi}\sigma} \exp\left(-\frac{(x - \sigma\sqrt{2}|\alpha| \cos(\omega t - \delta))^2}{\sigma^2}\right)$$

and the time dependent expectation value of position is

$$\langle x \rangle = \sigma\sqrt{2}|\alpha| \cos(\omega t - \delta).$$

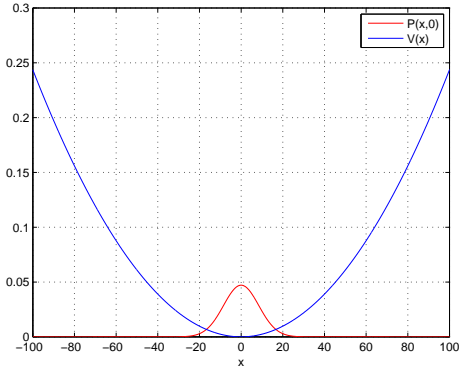


Figure 5.5:  $P(x, 0)$  and  $V(x)$

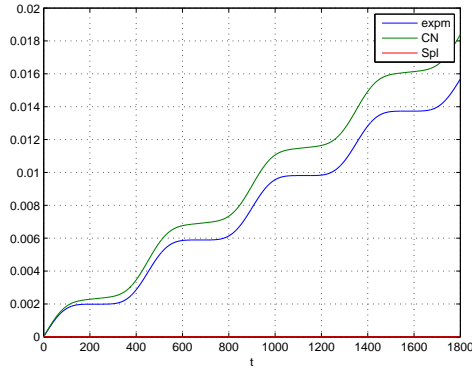


Figure 5.6:  $d(t)$  vs.  $t$  for the three methods

Again, we observe that the operator splitting method delivers more accurate solutions than the Crank-Nicholson scheme or the method of scaling and squaring does. These last two methods show similar behavior because they belong to the same class of numerical Integrators.

In Figure (5.7) we can see that the symmetric operator splitting scheme is not energy conserving whereas both the Crank-Nicholson scheme and the method of scaling and squaring are. However, we also observe that at least in the case of the harmonic oscillator the error in terms of energy remains small. As we want to find solutions of the Schrödinger equation for large times  $T$ , the symmetric splitting operator scheme seems to be a good candidate for this job. Although it does not conserve energy it may be possible to enclose the error by small constants.

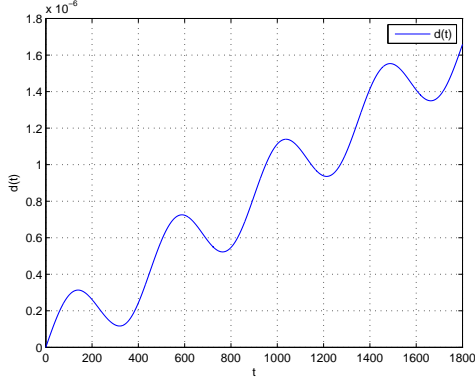


Figure 5.7: Error function  $d(t)$  in case of  $\Phi_{spl}(\tau)$

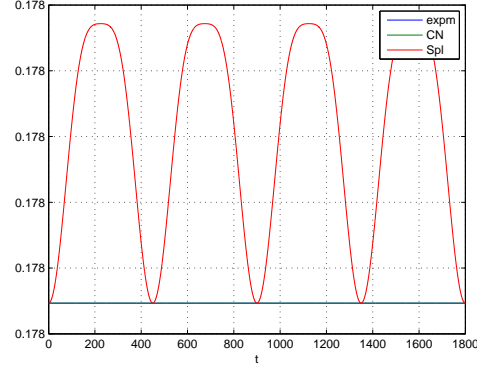


Figure 5.8: Energy vs. time for the three methods

### 5.3 A random potential

An interesting setup emerges if we model the multiplication operator  $V_\omega(x)$  as a random quantity itself. The resulting Hamiltonian  $H_\omega = H_0 + V_\omega(x)$  is a well known object in mathematical physics. These, so called random Schrödinger operators, model disordered solids. Depending on the physical model  $V_\omega(x)$  has different forms. In our case we choose an alloy-type potential  $V_\omega(x)$  as a random gaussian field in  $[-L, L]$ ,  $L > 0$ , given by

$$V_\omega(x) = \sqrt{\delta} \sum_{j=-L/\delta}^{L/\delta} \eta_j(\omega) q(x - j\delta).$$

Alloy type potentials model solids consisting of different materials. Whereas the distance function  $q(x)$  measures the distance between a particle, e.g. electron, at current position  $x$  and an atom at the lattice point  $j\delta$ , the type of atoms are unknown and therefore assumed to be random. Here we assume that the  $\eta_j$  are independent, uncorrelated and identically distributed random variables with expectation value  $\mathbb{E}(\eta_j) = 0$ , and variance  $\mathbb{V}ar(\eta_j) = 1$ . Defined in this way  $V_\omega(x)$  is a stationary stochastic process. We use Matlabs built in function `randn`, to find realizations of  $\omega_j$ . This routine returns a pseudorandom scalar value drawn from the normal distribution with zero mean and variance of one.

In order to investigate how reliable our numerical integrators are, lets denote  $\psi_t(\Phi(\tau))$  as the solution of Schrödinger equation at time  $t$  computed by the method  $\Phi(\tau)$  with the time sampling width  $\tau$ . Then a method  $\Phi(\tau)$  is reliable if it is continuous in the sense that if the perturbation of  $\tau$  is small, say  $\tau + \epsilon$ , then  $\|\psi_t(\Phi(\tau)) - \psi_t(\Phi(\tau + \epsilon))\|$  is also small. Therefore we define the function  $e(t) = \|\psi_t(\Phi(\tau_1)) - \psi_t(\Phi(\tau_2))\|$  which measures the error at time  $t$ . Though the sampling width with respect to the spatial variable does not affect the accuracy of the symmetric operator splitting method, it has

an effect on the computed solutions insofar that the resolution of the potential  $V(x)$  is high if  $h$  is small, with other words  $V(x)$  has a short scale structure of order  $h$ . For the following test phases we fix

$$\psi(x, 0) = \frac{1}{\sqrt{\sigma}\sqrt{\pi}} \exp(ix) \exp\left(\frac{(x - x_0)^2}{2\sigma^2}\right)$$

with  $\sigma = L/10$  and  $x_0 = -L + \sigma$  as initial condition.

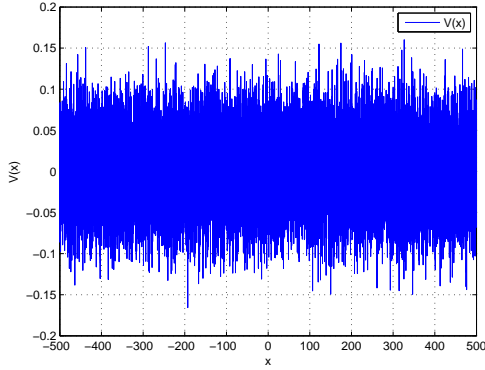


Figure 5.9:  $V(x)$  with  $h = \pi/50$

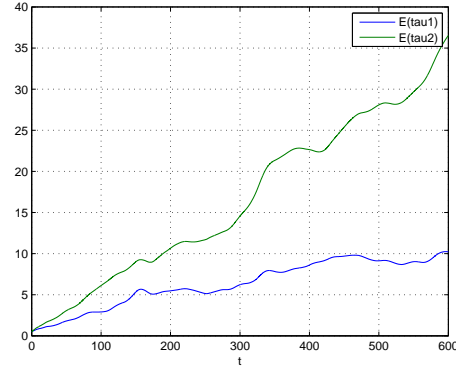


Figure 5.10:  $E(\tau_1)$  and  $E(\tau_2)$

In Figure (5.10) we see that the energy  $E(\tau_1)$  and  $E(\tau_2)$  for  $\tau_1 = 1/10$  and  $\tau_2 = 1/20$  increases with time, and that the error growth is linear in time, which confirms the analysis above. Theorem (2.3) shows that the error in terms of energy is of order  $\mathcal{O}(\tau^2)$  locally, but unfortunately does not shed light on the rate of error accumulation.

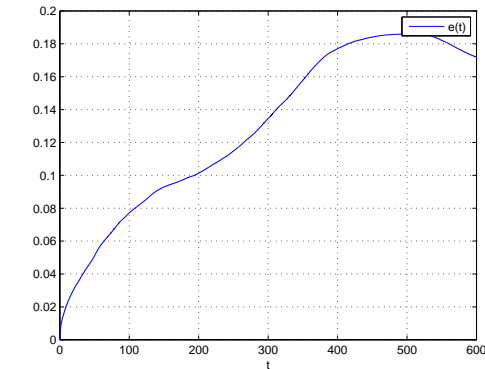


Figure 5.11:  $e(t)$  for  $T = 600$ ,  
 $\tau_1 = 1/10$  and  $\tau_2 = 1/20$

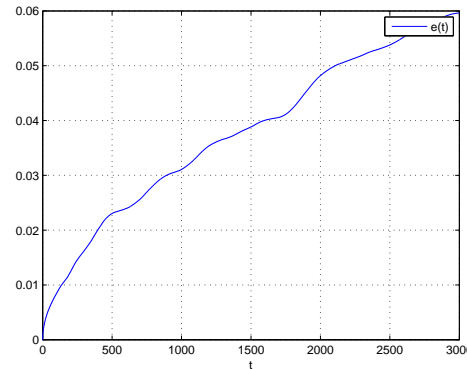


Figure 5.12:  $e(t)$  for  $T = 3000$ ,  
 $\tau_1 = 1/20$  and  $\tau_2 = 1/40$

Figures (5.11) and (5.12) show that the operator splitting method is reliable. However, the error in terms of energy grows too fast even if the step size  $\tau$  is small. Both, the

energy of the particle and the error accumulation with respect to energy during computation depend on the initial data. Basically it is valid: The smaller one chooses the velocity of the particle, the slower  $E(\tau)$  grows. In this case one can choose a bigger total time  $T$  which results in a smaller coupling constant  $\lambda$ . On the other hand, if we fix the velocity of the wave packet but enlarge its width  $\sigma$ , we have to adapt  $T$  to ensure that the wave packet does not hit the boundary. In this context the  $E(\tau)$  grows even faster. However, in both cases a variation of the initial data has not a satisfying effect on  $E(\tau)$ . To find the reason for that we test two further potentials.

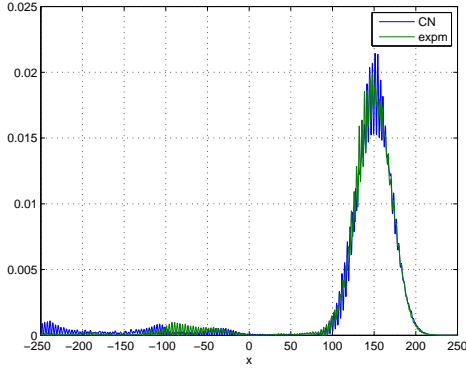


Figure 5.13:  $\psi(x, T = 300)$

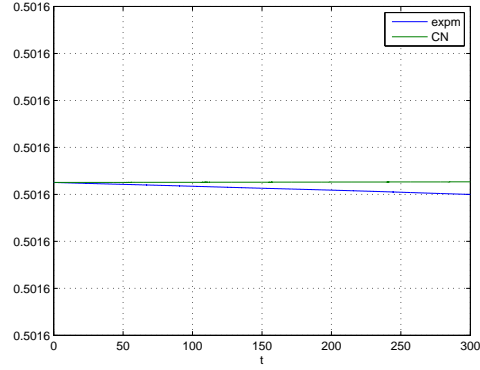


Figure 5.14:  $E(t)$  comp. by  $\Phi_{CN}(\tau)$  and  $\Phi_{\tau}$

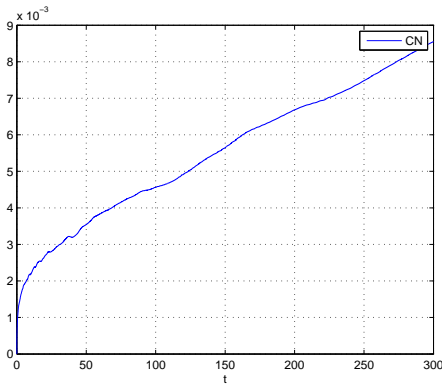


Figure 5.15:  $e(t)$  comp. with  $\Phi_{CN}(\tau)$

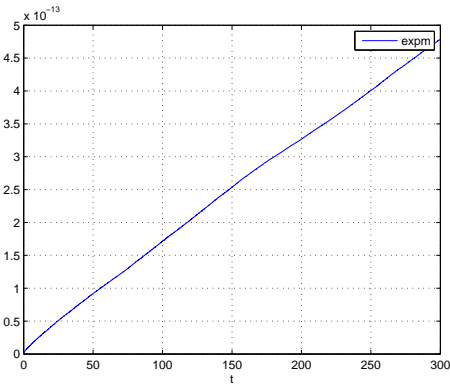


Figure 5.16:  $e(t)$  comp. with  $\Phi_{Sas}(\tau)$

The figures (5.13 - 5.16) show the same test run with the  $\Phi_{CN}$  and  $\Phi_{pad}$ . We can observe that the accuracy here is higher and energy conserved.

In order to find the reason why the symmetric operator splitting method failed in the case of the random potential, we test two further type of potentials.

## 5.4 A Mexican hat potential

This potential is similar to the harmonic oscillator potential. It is given by

$$V(x) = \left( x^2 - \left( \frac{L}{2} \right)^2 \right)^2$$

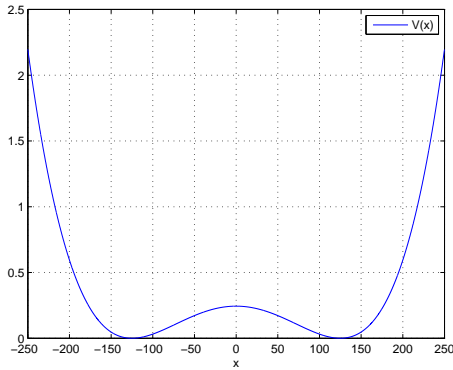


Figure 5.17: Mexican hat potential

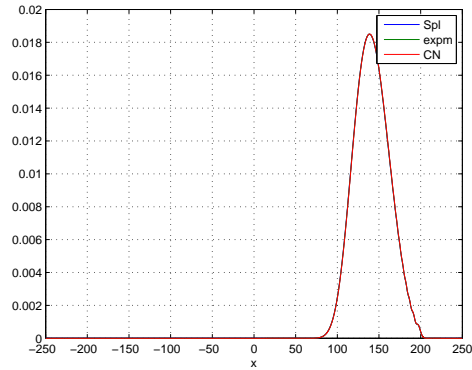


Figure 5.18:  $\psi(x, T = 300)$

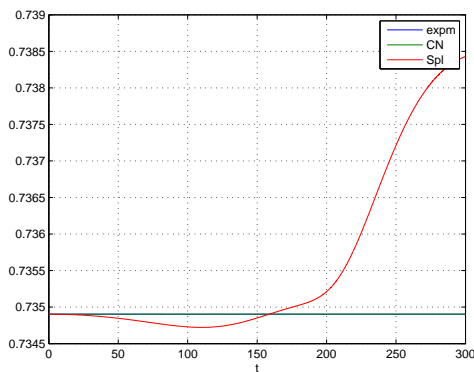


Figure 5.19:  $E(t)$  computed with  $\Phi_{CN}(\tau)$ ,  $\Phi_{Sas}(\tau)$  and  $\Phi_{Spl}(\tau)$

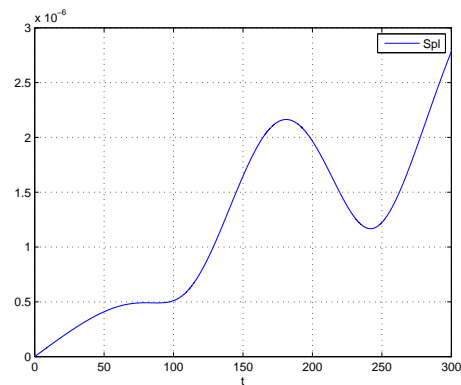


Figure 5.20:  $e(t)$  computed with  $\Phi_{Spl}(\tau)$

As expected all of the three methods show good approximation properties in this case. The reliability test for the scaling and squaring methods is very good, and the reliability test for the operator splitting method is even better than the one for the Crank-Nicholson method. Basically we can observe the same we did in the harmonic oscillator case. In figure (5.21) we can observe that  $E(\tau)$  grows slowly.

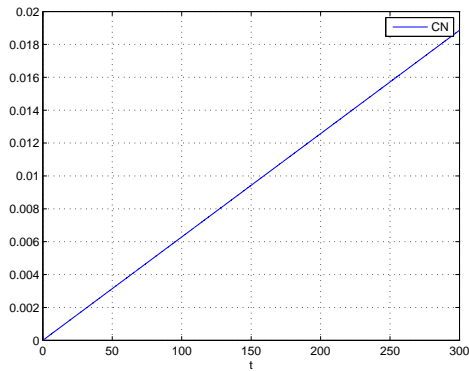


Figure 5.21:  $e(t)$  computed with  $\Phi_{Spl}(\tau)$

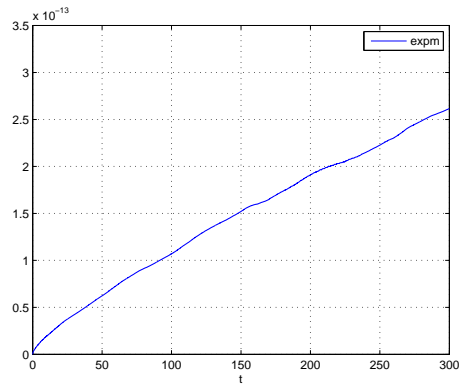


Figure 5.22:  $e(t)$  computed with  $\Phi_{Sas}(\tau)$

## 5.5 An Oscillating potential

Now we replace the potential with

$$V(x) = \left(\frac{x}{L}\right)^2 + \sin(\sqrt{2}x)$$

which has a short scale structur.

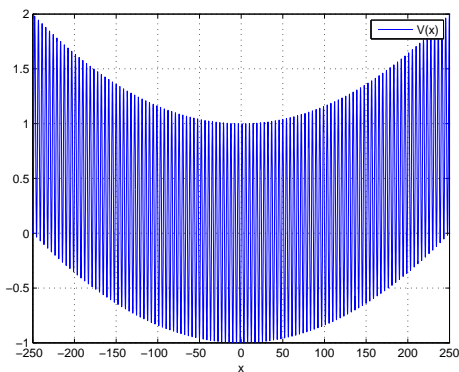


Figure 5.23: Potential  $V(x)$

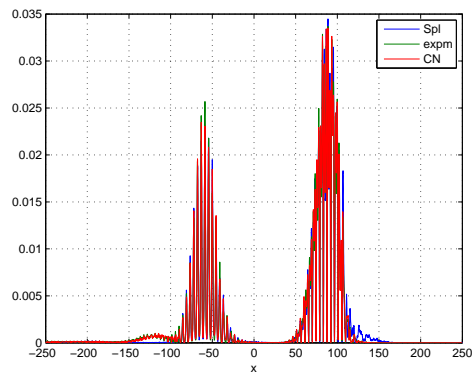


Figure 5.24:  $\psi(x, T = 300)$

As we can see in figure (5.24), the solution computed by the three methods are alike. In figure (5.25) we can observe how the energy related error grows. As before it is bounded by a constant of order  $\tau$ . The figures (5.26)-(5.28) show that all of the methods are reliable in this case, in particular the method of scaling and squaring.

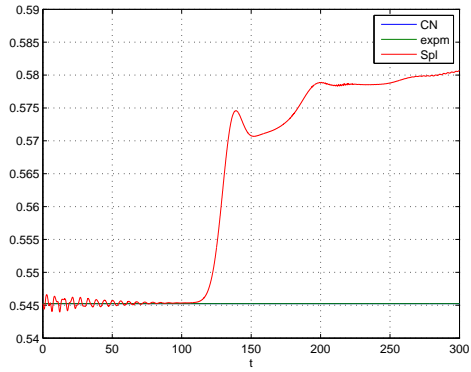


Figure 5.25:  $E(\tau)$  computed with  $\Phi_{CN}(\tau)$ ,  $\Phi_{Sas}(\tau)$  and  $\Phi_{Spl}(\tau)$

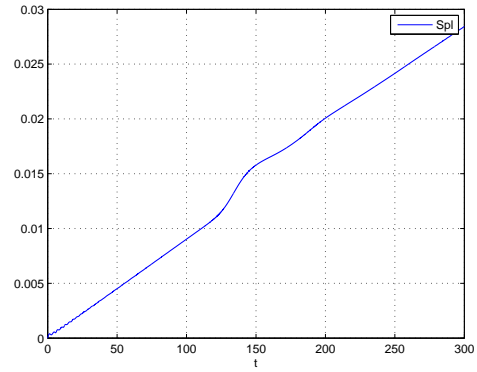


Figure 5.26:  $e(t)$  computed with  $\Phi_{Spl}(\tau)$

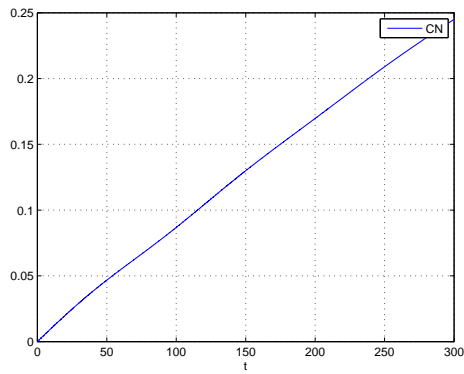


Figure 5.27:  $e(t)$  computed with  $\Phi_{CN}(\tau)$

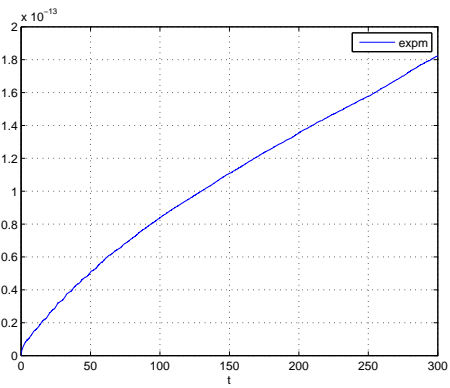


Figure 5.28:  $e(t)$  computed with  $\Phi_{Sas}(\tau)$

		$V(x) = 0$	$V(x) = \frac{1}{2}\omega^2 x^2$	$V(x) = (x^2 - \frac{L}{2})^2$	$V(x) = (\frac{x}{L})^2 + \sin(x)$	$V(x) = V_\omega(x)$
Energy	CN	conserved	conserved	conserved	conserved	conserved
	Spl	conserved	n.c., small error	n.c., small error	n.c, small error	n.c., large error
	Sas	conserved	conserved	conserved	conserved	conserved
Consistency	CN	$\max d(t)$ small	$\max d(t)$ small	—	—	—
	Spl	$\max d(t)$ tiny	$\max d(t)$ tiny	—	—	—
	Sas	$\max d(t)$ small	$\max d(t)$ small	—	—	—
Reliability	CN	—	—	$\max e(t)$ small	$\max d(t)$ small	$\max d(t)$ small
	Spl	—	—	$\max d(t)$ small	$\max d(t)$ small	$\max d(t)$ small
	Sas	—	—	$\max d(t)$ tiny	$\max d(t)$ tiny	$\max d(t)$ tiny

## 6 Wigner quasi probability distribution

The Wigner quasi probability distribution for a given function  $\psi$  in dimension  $d = 1$  is defined by

$$W_\psi(x, k) = \frac{1}{2\pi\hbar} \int_{-\infty}^{\infty} \psi^* \left( x + \frac{y}{2} \right) \psi \left( x - \frac{y}{2} \right) \exp(iky) dy,$$

where asterisk denotes the complex conjugate. We now derive a discrete Version of the Wigner function.

If we fix an  $x_j \in X$  and rewrite  $W_\psi(x, k)$  as

$$W_\psi(x_j, k) = \frac{1}{2\pi\hbar} \int_{-\infty}^{\infty} f_{x_j}(y) \exp(iky) dy \quad (6.1)$$

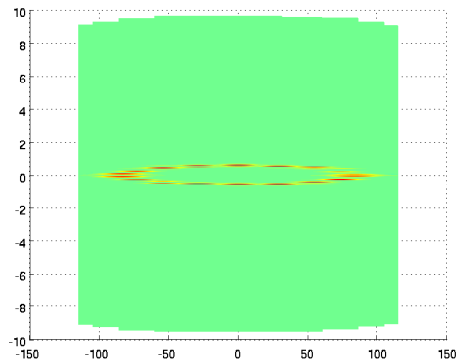
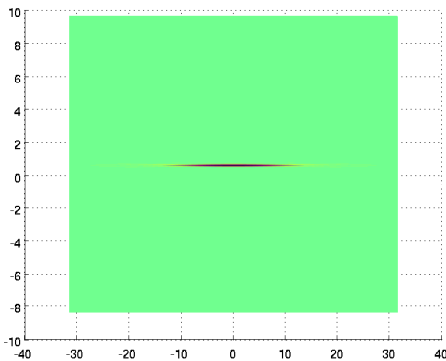
with  $f_{x_j}(y) = \psi^* \left( x + \frac{y}{2} \right) \psi \left( x - \frac{y}{2} \right)$ , we observe that (6.1) is the inverse Fourier transform of  $f_{x_j}(y)$ . Therefore it remains to find a way to calculate  $f_{x_j}(y)$ . Among the nice properties of the Fourier transform one presents itself very prominent here. So we utilize the shift theorem to calculate

$$\psi(x_j + y) \quad 1 \leq j \leq m.$$

and

$$\psi(x_j - y) \quad 1 \leq j \leq m.$$

If we choose  $y_j = \frac{x_j}{2}$  for  $1 \leq j \leq m$  then  $\psi(x_j + y_s)$  and  $\psi(x_j - y_s)$ ,  $1 \leq j, s \leq m$ , are  $(m, m)$ -Matrices calculated via the fast Fourier transform algorithm (fft). From these two matrices we can form the matrix  $f_{x_j}(y_s)$ ,  $1 \leq j, s \leq m$ , and gain  $W_\psi(x, k)$  by performing the integration via fft.



## 7 Conclusion

In chapter two we have seen that the symmetric operator splitting method does not conserve energy. However, since the energy related error is linear in  $\tau$  we received good results for the harmonic oscillator case. For the mexican hat potential, for which the time evolution of the Schrödinger equation can not be solved analytically, the splitting method is still reliable and conserves energy up to order  $\tau$ . Even if we replace the potential by a fast oscillating function, the operator splitting method remains reliable and produces solutions comparable with the solutions computed by the Crank-Nicholson method. Thus we follow that the error produced by the operator splitting method in the case of a random potential term does not depend on the short scale structure of the potential  $V(x)$ . One can suspect that the error produced by the splitting method depends on the differentiability of the potential.

Both, the method of scaling and squaring and the Crank-Nicholson method produce accurate solutions and conserves energy. However, due to the direct discretisation of the Laplacian, these methods are improper for numerical computation of the long time evolution of the Schrödinger equation.

## 8 Acknowledgment

First of all, I thank my family who supported me kindly the last month. I deeply thank Prof. László Erdős for providing this Diploma thesis and for his patient and kindly mentoring. I also thank Claudia Warnt and Florian Ranzi for reading the thesis and the invaluable feedback.

## 9 Appendix

### 9.1 Dft

The discrete Fourier transform (dft) is the Fourier transform applied to a discrete periodic function. An algorithm computing the dft at low computational cost was published by J.W. Cooley and John Tukey in the early 1970s. This popular method computes the dft of a sequence of length  $N$  by a computational complexity of  $N \log_2 N$ . We may assume an interval  $[0, L]$ ,  $x_j = jh$  and  $f_j = f(x_j)$ ,  $0 \leq j \leq m - 1$  are given. Then for given small  $h$  we can try to approximate

$$g(k) = \int_{-\infty}^{\infty} \exp(-2\pi i k x) f(x) dx \approx h \sum_{j=1}^m \exp(-2\pi i k x_j) f_j := \hat{f}(k)$$

Since for any  $z \in \mathbb{Z}$  it is valid that

$$-i \left( k + \frac{2\pi}{h} z \right) = -i k x_j - i 2\pi j z,$$

$\hat{f}(k)$  is a  $\frac{2\pi}{h}$  periodic function. We set  $2k_c = \frac{2\pi}{h}$  and let  $k \in [-k_c, k_c]$ . If now  $\text{supp } g(k) \subset [-k_c, k_c]$ ,  $\hat{f}(k)$  is a good approximation of  $g(k)$ .  $k_c$  is called critical frequency, also know as Nyquist frequency. If one chooses  $k$  bigger than  $k_c$ , say  $k = k_c + \epsilon$ ,  $\epsilon > 0$  then high oscillations cannot be reconstructed by the position space discretisation. In this case the function oscillates like  $\cos((k_c - \epsilon)x)$  at the mesh points. This phenomena is called aliasing. In order to avoid aliasing we have to choose the momentum space discretisation with care. Given a wave function with wave number  $p/\hbar \in [-k_c, k_c]$  we can utilize the periodicity of  $\hat{f}(k)$  and define

$$\begin{aligned} k_p &:= p\nu & p &= 0 \dots \left[ \frac{m+1}{2} \right] - 1 \\ k_q &:= q\nu - 2k_c & q &= \left[ \frac{m+1}{2} \right] \dots m - 1, \\ k &= (k_p, k_q) \end{aligned}$$

where  $[a]$  means to round  $a$  down to the next integer. If we choose  $\nu$  as

$$\nu := \frac{2k_c}{N} = \frac{2\pi}{mh} = \frac{2\pi}{L}$$

the discrete fourier tranform of a vector  $v$  is given by

$$\hat{v}_{k+1} = (\mathcal{F}v)_{k+1} = \sum_{j=0}^{m-1} v_{j+1} e^{-2\pi i j k / m}.$$

And its inverse transform

$$v_{j+1} = (\mathcal{F}^{-1}v)_{j+1} = \frac{1}{m} \sum_{k=0}^{m-1} \hat{v}_{k+1} e^{2\pi i j k / m}$$

where  $i = \sqrt{-1}$  and  $k = 0, \dots, m-1$ .

Parsevals theorem in the discrete case reads

$$\sum_{j=0}^{m-1} |f_j|^2 = \frac{1}{m} \sum_{k=0}^{m-1} |\hat{f}_k|^2.$$

Therefore the map  $v \rightarrow \frac{1}{\sqrt{(m)}}(\mathcal{F}v)$  is unitary. Note that the definitions of the Fourier transform above are consistent with Matlab's definitions of the discrete Fourier transform. However, Matlab's implementations of the fast fourier tranforms does not fulfill the Parseval Theorem.

## 9.2 Quadrature formulae

Let  $Q = (b_j, c_j)_{j=1}^m$  be a quadrature formula for  $f \in C[a, b]$  and  $v = (z_0, \dots, z_N)$  a decomposition of the interval  $[a, b]$ . Then

$$I[f] := \int_a^b f(x) dx = \sum_{i=1}^N \int_{z_{i-1}}^{z_i} f_i(x) dx \approx \sum_{i=1}^N Q[f_i]$$

where  $f_i : [z_{i-1}, z_i] \rightarrow \mathbb{R}$  with  $f_i(x) = f(x)$ . For a given function  $f : [x_0, x_0 + h] \rightarrow \mathbb{R}$  we define  $s : [0, 1] \rightarrow [x_0, x_0 + h]$ ,  $s(t) = x_0 + th$ . Then by substitution rule it is valid

$$\begin{aligned} \int_{x_0}^{x_0+h} f(x) dx - Q[f] &= \int_0^1 f(s(t)) s'(t) dt - Q[f] \\ &= h \left( \int_0^1 f(s(t)) dt - \sum_{j=1}^s b_j f(x_0 + c_j h) \right) \end{aligned}$$

and therefore with  $g : [0, 1] \rightarrow \mathbb{R}$ ,  $g(t) := f(s(t))$  it follows

$$\int_x^{x+h} f(x) dx - Q[f] = h \left( \int_0^1 g(t) dt - \sum_{j=1}^m b_j g(c_j) \right).$$

Thus, for a given quadrature formula it is sufficient to analyze the error

$$E[g] = \int_0^1 g(t)dt - \sum_{j=1}^m b_j g(c_j)$$

for continuous functions  $g : [0, 1] \rightarrow \mathbb{R}$ . A nice result in this direction is the Peano kernel-theorem.

```

0001 function main
0002
0003
0004 T = input('Please enter total physical time T: ');
0005 tau = input('Please enter time step size: ');
0006
0007 t=(0:tau:T)';
0008
0009
0010 %x-space
0011 L = input('Please enter spatial length L: ');
0012 h = input('Please enter spatial step size: ');
0013 x = (-L:h:L)';
0014
0015 disp('-----')
0016 disp('0 Free space test.')
0017 disp('1 Harmonic oscillator test.')
0018 disp('2 Mexican hat potential test.')
0019 disp('3 Short scale structure potential test.')
0020 disp('4 Random potential test')
0021 disp('-----')
0022
0023 mode = input('Type 0, 1, 2, 3 or 4: ');
0024
0025 tau2 = tau/2;
0026 % Laplacian
0027 m=length(x);
0028 A = eye(m)*(2) + diag(-ones(m-1,1),-1) + diag(-ones(m-1,1),1);
0029 c=(1/(2*h^2));
0030 A=c*A;
0031 delta = 1/10;
0032
0033 omega = 2*pi/900;
0034
0035 if(mode==1)
0036     v = ((omega*x).^2)/2;
0037 else
0038     if(mode==2)
0039         v = (x.^2-(L/2)^2).^2;           % Mexican hat potential
0040     else
0041         if(mode==3)
0042             v = (x/L).^2 +sin(x);       % Potential with short scale structure.
0043         else
0044             if(mode==4)
0045                 testv(x,delta,L);
0046                 v = load('testpotential.mat');
0047                 v=(1/sqrt(T))*(v.v);    % random gaussian field potential
0048             else
0049                 v = 0;                  % The Hamiltonian for the free space
0050             end
0051         end
0052     end
0053 end
0054 A = A + diag(v);                       % The Hamiltonian for the harmonic oscillator
0055
0056 % scaling and squaring method
0057 [psi1,diff1,en1] = pade_method(L,x,t,tau,tau2,h,A,mode);
0058
0059 % Crank-Nicholson method
0060 [psi2,diff2,en2] = cn_method(L,x,t,tau,tau2,h,A,mode);
0061
0062 % splitting method
0063 [psi3,diff3,en3] = spl_method(L,x,t,tau,tau2,h,v,A,mode);
0064
0065
0066 % Plot of the wave packet at time T computed with the different methods
0067 f1=figure(1);
0068 plot(x,psi1.*conj(psi1),x,psi2.*conj(psi2),x,psi3.*conj(psi3));
0069 title('wave packet')
0070 xlabel('x');
0071 legend('expm','CN','Spl');
0072 saveas(f1,'all.fig');
0073
0074 % Plot of the function d(t)
0075 f2=figure(2);
0076 plot(t,diff1,t,diff2,t,diff3);
0077 title('reliability resp. error')
0078 legend('expm','CN','Spl');
0079 saveas(f2,'diff.fig');
0080
0081 % Plot of the energy
0082 f3=figure(3);
0083 plot(t,en1,t,en2,t,en3);
0084 title('energy')
0085 legend('expm','CN','Spl');
0086 saveas(f3,'Energy.fig');

```

```
0087
0088 % For better illustration: Plot of d(t) in the case of the symmetric
0089 % operator splitting method
0090 f4=figure(4);
0091 plot(t,diff3);
0092 title('d(t) for the splitting method')
0093 legend('Spl');
0094 saveas(f4,'diff2.fig');
0095
0096
0097 f5 = figure(5);
0098 subplot(2,2,1)
0099 plot(x,v);
0100 l1=legend('V(x)');
0101 set(l1,'FontSize',6);
0102 axis tight;
0103 title('Potential')
0104 subplot(2,2,2)
0105 plot(x,psi1.*conj(psi1),x,psi2.*conj(psi2),x,psi3.*conj(psi3));
0106 l2=legend('expm', 'CN', 'Spl');
0107 set(l2,'FontSize',6);
0108 title('wave packet')
0109 subplot(2,2,3)
0110 plot(t,diff1,t,diff2,t,diff3);
0111 l3=legend('expm', 'CN', 'Spl');
0112 set(l3,'FontSize',6);
0113 title('reliability resp. error')
0114 subplot(2,2,4)
0115 plot(t,en1,t,en2,t,en3);
0116 l4 = legend('expm', 'CN', 'Spl');
0117 set(l4,'FontSize',6);
0118 title('energy')
0119 saveas(f5,'allinone.fig');
```

```

0001 function [v1,v2,v3] = cn_method(L,x,t,tau,tau2,h,H,mode)
0002
0003 m=length(x);
0004
0005 if(mode==2 || mode==3 || mode==4)
0006     CN2 = inv(2*eye(m)+i*tau2*H)*(2*eye(m)-i*tau2*H);
0007
0008     sigma = L/10;
0009     x0 = 0;
0010     psi2=(1/(sqrt(sigma*sqrt(pi))))*exp(i*x).*exp(-((x-x0).^2)/(2*sigma^2));
0011 end
0012
0013 % propagator
0014 CN = inv(2*eye(m)+i*tau*H)*(2*eye(m)-i*tau*H);
0015
0016 % initial condition
0017 if(mode==1)
0018     psi = haos(0,x);
0019 else
0020     psi=free_space(0,x);
0021 end
0022
0023 diff = zeros(1,length(t));
0024 energy = zeros(1,length(t));
0025
0026 if(mode==2 || mode==3 || mode==4)
0027     psi = psi2;
0028     wb = waitbar(0,'Processing with CN-Method');
0029     for j=1:length(t)
0030
0031         % difference in the L2-norm
0032         diff(1,j)=sqrt(h*sum((psi2-psi).*conj(psi2-psi)));
0033
0034         % energy
0035         energy(1,j)=h*(sum(psi.*conj(H*psi)));
0036
0037         % one time step propagation
0038         psi=CN*psi;
0039         psi2 = CN2*(CN2*psi2);
0040
0041         waitbar(j/length(t));
0042     end
0043     close(wb);
0044 end
0045
0046 wb = waitbar(0,'Processing with CN-Method');
0047 if(mode==1)
0048     for j=1:length(t)
0049         % exact solution
0050         h1 = haos(t(j),x);
0051
0052         % error in the L2-norm
0053         diff(1,j)=sqrt(h*sum((h1-psi).*conj(h1-psi)));
0054
0055         % energy
0056         energy(1,j)=h*(sum(psi.*conj(H*psi)));
0057
0058         % one time step propagation
0059         psi=CN*psi;
0060
0061         waitbar(j/length(t));
0062     end
0063     close(wb);
0064 else if(mode==0)
0065     for j=1:length(t)
0066
0067         % exact solution
0068         h1 = free_space(t(j),x);
0069
0070         % error in the L2-norm
0071         diff(1,j)=sqrt(h*sum((h1-psi).*conj(h1-psi)));
0072
0073         % energy
0074         energy(1,j)=h*(sum(psi.*conj(H*psi)));
0075
0076         % one time step propagation
0077         psi=CN*psi;
0078
0079         waitbar(j/length(t));
0080     end
0081 end
0082 close(wb);
0083 end
0084
0085 v1=psi;
0086 v2=real(diff);
0087 v3=real(energy);

```

```

0001 function [v1, v2, v3] = pade_method(L,x,t,tau,tau2,h,H,mode)
0002
0003 if(mode==2 || mode==3 || mode==4)
0004     E2= expm(-i*tau2*H);
0005
0006     sigma = L/10;
0007     x0 = 0;
0008     psi2=(1/(sqrt(sigma*sqrt(pi))))*exp(i*x).*exp(-((x-x0).^2)/(2*sigma^2));
0009 end
0010
0011
0012
0013 E=expm(-i*tau*H);
0014
0015 if(mode==1)
0016     psi = haos(0,x);
0017 else
0018     psi=free_space(0,x);
0019 end
0020
0021 diff = zeros(1,length(t));
0022 energy = zeros(1,length(t));
0023
0024 if(mode==2 || mode==3 || mode==4)
0025     psi = psi2;
0026     wb = waitbar(0, 'Processing with Scaling and Squaring method');
0027     for j=1:length(t)
0028
0029         % difference in the L2-norm
0030         diff(1,j)=sqrt(h*sum((psi2-psi).*conj(psi2-psi)));
0031
0032         % energy
0033         energy(1,j)=h*(sum(psi.*conj(H*psi)));
0034
0035         % one time step propagation
0036         psi=E*psi;
0037         psi2 = E2*(E2*psi2);
0038
0039         waitbar(j/length(t));
0040     end
0041 close(wb);
0042 end
0043
0044
0045
0046 wb = waitbar(0, 'Processing with Scaling and Squaring method');
0047 if(mode==1)
0048     for j=1:length(t)
0049
0050         % exact solution
0051         h1 = haos(t(j),x);
0052
0053         % error in the L2-norm
0054         diff(1,j)=sqrt(h*sum((h1-psi).*conj(h1-psi)));
0055
0056         % energy
0057         energy(1,j)=h*(sum(psi.*conj(H*psi)));
0058
0059         % one time step propagation
0060         psi=E*psi;
0061
0062         waitbar(j/length(t));
0063     end
0064 close(wb);
0065 else if(mode==0)
0066     for j=1:length(t)
0067         % exact solution
0068         h1 = free_space(t(j),x);
0069
0070         % error in the L2-norm
0071         diff(1,j)=sqrt(h*sum((h1-psi).*conj(h1-psi)));
0072
0073         % energy
0074         energy(1,j)=h*(sum(psi.*conj(H*psi)));
0075
0076         % one time step propagation
0077         psi=E*psi;
0078
0079         waitbar(j/length(t));
0080     end
0081 end
0082 close(wb);
0083 end
0084
0085 v1=psi;
0086 v2=real(diff);
0087 v3=real(energy);

```

```

0001 function [v1,v2,v3]=spl_method(L,x,t,tau,tau2,h,v,H,mode)
0002
0003 m=length(x);
0004
0005
0006 %K-space
0007 kc = pi/h;
0008 kh = 2*pi/(2*L);
0009 k(1,1:floor((m+1)/2)) = (0:floor((m+1)/2)-1)*kh;
0010 k(1,1:floor((m+1)/2)+1:m) = (floor((m+1)/2):m-1)*kh -2*kc;
0011
0012 % initial condition
0013 if(mode==1)
0014     psi = haos(0,x);
0015 else
0016     psi=free_space(0,x);
0017 end
0018
0019 if(mode==2 || mode==3 || mode==4)
0020     E2=exp(-i*(tau2/2)*(k.^2));
0021     V2 = exp(-i*(tau2/2)*v);
0022
0023     sigma = L/10;
0024     x0 = 0;
0025     psi2=1/(sqrt(sigma*sqrt(pi)))*exp(i*x).*exp(-(x-x0).^2)/(2*sigma^2));
0026 end
0027
0028 % propagator
0029 E = exp(-i*(tau/2)*(k.^2));
0030 V = exp(-i*(tau/2)*v);
0031
0032 diff=zeros(1,length(t));
0033 energy = zeros(1,length(t));
0034
0035
0036 if(mode==2 || mode==3 || mode==4)
0037     psi = psi2;
0038     wb = waitbar(0,'Processing with Symmetric Splitting method');
0039     for j=1:length(t)
0040
0041         % difference in the L2-norm
0042         diff(1,j)=sqrt(h*sum((psi2-psi).*conj(psi2-psi)));
0043
0044         % energy
0045         energy(1,j)=h*(sum(psi.*conj(H*psi)));
0046
0047         % one time step propagation
0048         psi = (V.*(ifft(E.*fft(V.*psi))));
0049         psi2 = (V2.*(ifft(E2.*fft(V2.*psi2))));
0050         psi2 = (V2.*(ifft(E2.*fft(V2.*psi2))));
0051
0052         waitbar(j/length(t));
0053     end
0054     close(wb);
0055 end
0056
0057
0058 wb = waitbar(0,'Processing with Symmetric Splitting method');
0059 if(mode==1)
0060     for j=1:length(t)
0061
0062         % exact solution
0063         h1=haos(t(j),x);
0064
0065         % error in the L2 norm
0066         diff(1,j)=sqrt(h*sum((h1-psi).*conj(h1-psi)));
0067
0068         % energy
0069         energy(1,j) = h*(sum(psi.*conj(H*psi)));
0070
0071         % one time step propagation
0072         psi = (V.*(ifft(E.*fft(V.*psi))));
0073
0074         waitbar(j/length(t));
0075     end
0076     close(wb);
0077 else if(mode==0)
0078
0079     for j=1:length(t)
0080
0081         % exact solution
0082         h1=free_space(t(j),x);
0083
0084         % error in the L2-norm
0085         diff(1,j)=sqrt(h*sum((h1-psi).*conj(h1-psi)));
0086
0087         % energy
0088         energy(1,j) = h*(sum(psi.*conj(H*psi)));

```

```
0089
0090     % one time step propagation
0091     psi = (V.*(ifft(E.*(fft(V.*psi)))));
0092
0093     waitbar(j/length(t));
0094     end
0095 end
0096 close(wb);
0097 end
0098
0099 v1=psi;
0100 v2=real(diff);
0101 v3=real(energy);
```

```
0001 function testv(x,delta,L)
0002
0003 m = length(x);
0004
0005 vm = zeros(m,1);
0006 for k = 1:m
0007     vm(k,1) = V(x(k),delta,L);
0008 end
0009
0010 save('testpotential', 'vm');
```

```
0001 function Vx = V(x,delta,L)
0002
0003
0004 j=(-L/delta:L/delta);
0005 jm = length(j);
0006
0007 r =randn(1,jm);
0008
0009 %Vx=r*exp(-(ones(jm,1)*x-j*delta*ones(1,m)).^2);
0010
0011
0012 Vx = sqrt(delta)*sum(r.*(exp(-(x-delta*j).^2)));
```

```

0001 function fspsi = free_space(t,x)
0002
0003
0004
0005
0006
0007 sigma0 = 5;           % Width of the Wavepacket
0008 k0=1;                 % velocity of the wave packet
0009 m=1;                  % unit mass
0010 hbar = 1;
0011 x0=50;                % initial localisation
0012 p0=hbar*k0;          % average momentum
0013
0014
0015 gn = 1/sqrt(sigma0*sqrt(pi)); % Normalisation
0016
0017
0018
0019 alpha = sqrt(sigma0^2 +i*hbar*t/m);
0020 C1 = sigma0/alpha;
0021
0022 % time evolution of the wave packet for V(x)=0
0023 fspsi = gn*C1*exp(i*k0*(x-p0*t/(2*m))).*exp((-x-x0-p0*t/m).^2)/(2*(sigma0^2 +i*hbar*t/m));

0001 function h1=haos(t,x)
0002
0003 alphab = 5;
0004 delta = 2*pi/4;
0005 alpha = alphab*exp(i*delta);
0006
0007
0008 omega = 2*pi/900;
0009 hbar = 1;
0010 m = 1;
0011
0012 x0 = sqrt(hbar/(omega*m));
0013
0014
0015 C = 1/(sqrt(sqrt(pi))*sqrt(x0));
0016
0017 %time evolution of a wave packet in a parabolic potential
0018 h1 = C*exp(-i*omega*t/2 - (alpha.*conj(alpha))*sin(2*(omega*t-delta))/2 + ...
... sqrt(2)*alphab*x*sin(omega*t-delta)/x0 - 1/(2*x0^2)*(x-x0*sqrt(2)*alphab*cos(omega*t-delta)).^2);

0001 function plot_wigner(v,x)
0002 % Plots the wigner quasi probability distribution
0003 % in a time-momentum system of koordinates.
0004 % In order to be able to compute the wigner distribution
0005 % for bigger arrays, data reduction is required.
0006
0007
0008 % Reduce the amount of data
0009 [xn kn pn] = dr(x,v,1000);
0010
0011 % Compute Wigner distribution
0012 w = wigner(pn,xn,kn);
0013
0014 % Restrict to relevant data
0015 [r,c] = find(w==max(w(:)));
0016 ran = 300;
0017 wu = w((r-ran):(r+ran),(c-ran):(c+ran));
0018
0019 % Adapt coordinates
0020 xu = xn((c-ran):(c+ran));
0021 kl = fftshift(kn);
0022 ku = kl((r-ran):(r+ran));
0023
0024 % Mesh plot with top view
0025 mesh(xu,ku,wu);
0026 view([0 90]);

```

```

0001 function W = wigner(psi,x,k)
0002 % Computes Wigners quasi probability distribution
0003 % for a given wave packet psi, an array of abscissae x and
0004 % an array of momentum koordinates k.
0005
0006 y = x/2;
0007
0008 m = length(x);
0009
0010
0011 k=fftshift(k);
0012
0013 fftpsi = (fft(psi));
0014
0015
0016 A = ((ifft((fftpsi*ones(1,m)).*(exp(i*k'*y')))));
0017 B = (ifft((fftpsi*ones(1,m)).*(exp(-i*k'*y'))));
0018
0019 f=(conj(A).*B)';
0020
0021 % Wigner quasi distribution function
0022 W = real(fft(f));

```

```

0001 function [xnew,knew, psinew] = dr(x,psi,range)
0002 % This function reduces the amount of data, for
0003 % a given wave function psi.
0004 % Most of the zero values are cleaned up from psi
0005 % which is better for further computation
0006
0007 % compute the index of the expectation value <x>
0008 [c1, ind] = max(psi.*conj(psi));
0009
0010
0011 i1 = ind - range;
0012 i2 = ind + range;
0013 m = length(x);
0014
0015 if(i1<0)
0016     i2 = i2 + abs(i1)+1;
0017     i1=1;
0018 else
0019     if(i2>m)
0020         i1=i1-(i2-m);
0021         i2 = m;
0022     end
0023 end
0024
0025
0026 xnew = x(i1:i2);
0027
0028 m2 = length(xnew);
0029 L = abs(x(i1)) +abs(x(i2));
0030
0031 h = x(2)-x(1);
0032
0033
0034 kc = pi/h;
0035 kh = 2*pi/L;
0036
0037 k(1,1:floor((m2+1)/2)) = (0:floor((m2+1)/2)-1)*kh;
0038 k(1, floor((m2+1)/2)+1:m2) = (floor((m2+1)/2):m2-1)*kh -2*kc;
0039
0040
0041 knew=k;
0042 psinew = psi(i1:i2);

```

# Bibliography

- [1] G. Evans, J. Blackledge, P. Yardley. Numerical Methods for Partial Differential Equations. Springer 2000
- [2] A. Garcia. Numerical Methods for Physics. Prentice-Hall 1994
- [3] Gene H. Golub, Charles F. Van Loan. Matrix Computations. THE JOHNS HOPKINS UNIVERSITY PRESS, Baltimore, Maryland 1983.
- [4] W. Gautschi, Numerical Analysis: An Introduction. Birkhäuser Boston, Boston, MA, 1997.
- [5] M.Hanke-Bourgeois. Grundlagen der numerischen Mathematik und des wissenschaftlichen Rechnens. Vieweg+Teubner Verlag, 2008.
- [6] A. Quarteroni, A. Valli. Numerical Approximation of Partial Differential Equations. Berlin, Springer 1997.
- [7] I. Rubinstein, L. Rubinstein. Partial Differential Equations in classical Mathematical Physics. Cambridge, Cambridge University Press 1998.
- [8] Quantenmechanik. Berlin, Springer 1988.
- [9] J. Weidmann. Linear Operators in Hilbert spaces. New York, Springer.
- [10] T. Jahnke and Ch. Lubich. Error bounds for exponential operator splittings. BIT 40/4 (2000), 735–744.
- [11] Nicholas J. Higham. The Scaling and Squaring Method for the Matrix Exponential Revisited. SIAM J. MATRIX ANAL. APPL. Vol. 26, No. 4, pp. 1179–1193
- [12] M. Suzuki. Fractal decomposition of exponential operators with applications to many-body theories and monte carlo simulations. Phys. Lett. A, 146:319–323, 1990.
- [13] H. Yoshida. Construction of higher order symplectic integrators. Phys. Lett. A, 150:262–268, 1990.