

Energiespektrum aus Autokorrelationsfunktion

Pavel Sterin

1 Split-Operator Methode

Die Split-Operator Methode ist ein Verfahren zur numerischen Approximation von Lösungen der Schrödingergleichung in kartesischen Koordinaten. Zur Vereinfachung der Berechnungen setze ich verwende ich die Konvention $\hbar = 1$, $m = \frac{1}{2}$. Damit erhält man eine einfache Form, der linearen DGL:

$$\begin{aligned} \left(-\frac{\partial^2}{\partial x^2} + V(x, t) \right) \psi(x, t) &= i \frac{\partial}{\partial t} \psi(x, t) \\ &= (T + V) \psi(x, t) = i \frac{\partial}{\partial t} \psi(x, t) \end{aligned}$$

Mit dem üblichen Ansatz für die zeitliche Evaluation von $\psi(x, t) = U(t)\psi(x)$ (mit fixierter Wellenfunktion $\psi(x)$ und unitärem Zeit-Propagator $U(t)$) erhält man die äquivalente Schrödingergleichung des Propagators:

$$(T + V(t)) U(t) = i \frac{\partial}{\partial t} U(t)$$

Die formale Lösung dieser Gleichung ist bekannterweise die Dyson Reihe, oder etwas weniger allgemein für verschiedene Zeiten mit sich selbst kommutierenden Hamiltonian $H(t) = T + V(t)$, $[H(t), H(t')] = 0$ ist:

$$U(t) = \exp \left(\int_0^t -i(T + V(t')) dt' \right)$$

Falls sich das Potential für kleine Zeitdifferenzen τ nur geringfügig ändert oder sogar stets konstant ist, sodass $[H(t), H(t + \tau)] \approx 0$ gilt, geht diese Lösung in das Operator-Exponential über:

$$U(t, \tau) = \exp(-i\tau(T + V(t)))$$

Im weiteren beziehe ich mich stets auf eine einzige Iteration und lasse darum die zeitliche Abhängigkeit von V fallen:

$$U(\tau) = e^{-i\tau(T+V)} \quad (1)$$

Da T nur im Impulsraum und V nur im Ortsraum wirken, wäre es günstig sie nur dort auszuwerten, weil sie dann als Multiplikationsoperatoren diagonal wären und das Exponential sich leicht auswerten ließe. Da jedoch für die meisten interessanten Probleme T und

V nicht kommutieren, $[T, V] \neq 0$, muss man hier geschickt Approximieren.

Für jeden Operator gilt zunächst:

$$e^{-i\tau X} = \sum_{n=0}^{\infty} \frac{(i\tau X)^n}{n!} = 1 - i\tau X - \tau^2 \frac{X^2}{2} + O(\tau^3)$$

Mit den Bezeichnung $U_X(\tau) = e^{-i\tau X}$ ergibt das dann:

$$\begin{aligned} U_{X+Y}(\tau) &= 1 - i\tau(X + Y) - \tau^2 \frac{(X + Y)^2}{2} + O(\tau^3) \\ &= 1 - i\tau X - i\tau Y \\ &\quad - \frac{\tau^2}{2} (X^2 + XY + YX + Y^2) + O(\tau^3) \end{aligned}$$

$$U_X(\tau) = 1 - i\tau X - \tau^2 \frac{X^2}{2} + O(\tau^3)$$

$$U_Y(\tau) = 1 - i\tau Y - \tau^2 \frac{Y^2}{2} + O(\tau^3)$$

$$\begin{aligned} U_X(\tau)U_Y(\tau) &= 1 - i\tau Y - i\tau X \\ &\quad - \tau^2 XY - \tau^2 \frac{Y^2}{2} - \tau^2 \frac{X^2}{2} + O(\tau^3) \end{aligned}$$

Damit erhält man den Ausdruck:

$$U_{X+Y}(\tau) = U_X(\tau)U_Y(\tau) + \frac{\tau^2}{2} [X, Y] + O(\tau^3) \quad (2)$$

Mit $X = \frac{V}{2}$, $Y = T + \frac{V}{2}$ folgt:

$$\begin{aligned} U_{\frac{V}{2}+T+\frac{V}{2}}(\tau) &= U_{\frac{V}{2}}(\tau)U_{T+\frac{V}{2}}(\tau) \\ &\quad + \left[\frac{V}{2}, T + \frac{V}{2} \right] + O(\tau^3) \\ &= U_{\frac{V}{2}}(\tau)U_{T+\frac{V}{2}}(\tau) + \left[\frac{V}{2}, T \right] + O(\tau^3) \\ &= U_{\frac{V}{2}}(\tau) \left(U_T(\tau)U_{\frac{V}{2}}(\tau) + \frac{\tau^2}{2} \left[T, \frac{V}{2} \right] \right) \\ &\quad + \left[\frac{V}{2}, T \right] + O(\tau^3) \\ &= U_{\frac{V}{2}}(\tau)U_T(\tau)U_{\frac{V}{2}}(\tau) + O(\tau^3) \quad (3) \end{aligned}$$

Durch die symmetrische Aufteilung des Potentials verringert sich also der Fehler auf die 3. Ordnung.

In dieser Darstellung ist jetzt einfach die Wirkung von $U \approx \tilde{U} = U_{\frac{V}{2}}(\tau)U_T(\tau)U_{\frac{V}{2}}(\tau)$ zu berechnen, denn:

$$U_{\frac{V}{2}}(\tau)\psi(x) = e^{-i\frac{\tau}{2}V(x)}\psi(x) \quad (4)$$

$$U_T(\tau)\psi(x) = \mathcal{F}^{-1}e^{-i\tau k^2}\mathcal{F}\psi(x) \quad (5)$$

wobei \mathcal{F} die Fouriertransformation ist.

Diese beiden Gleichungen definieren die Iteration der Split-Operator-Methode für n Schritte bzw. Zeit $t = n\tau$

- Start**
- Propagation von $\psi(x)$ mit $U_{\frac{V}{2}}(\tau)$
 - Fouriertransformation zu $\psi(k)$
 - Propagation von $\psi(k)$ mit $U_T(\tau)$

- Iteration**
- inverse Fouriertransformation zu $\psi(x)$
 - Propagation von $\psi(x)$ mit $U_{\frac{V}{2}}(\tau)$
 - Propagation von $\psi(x)$ mit $U_{\frac{V}{2}}(\tau)$
 - Fouriertransformation zu $\psi(k)$
 - Propagation von $\psi(k)$ mit $U_T(\tau)$

- Ende**
- inverse Fouriertransformation zu $\psi(x)$
 - Propagation von $\psi(x)$ mit $U_{\frac{V}{2}}(\tau)$

wobei die Iteration $(n - 1)$ Mal ausgeführt wird. Die zwei nacheinander folgenden Propagationen mit $U_{\frac{V}{2}}(\tau)$ werden in der Implementierung natürlich zu $U_V(\tau)$ zusammengefasst.

Bemerkung: Es hindert einen niemand daran die Rollen von T und V zu vertauschen und statt \tilde{U} den Operator $U_{\frac{T}{2}}(\tau)U_V(\tau)U_{\frac{T}{2}}(\tau)$ zu verwenden. Der einzige Unterschied ist eine zusätzliche Fouriertransformation am Anfang und eine inverse Fouriertransformation am Ende des Algorithmus, die mir angesichts der eher kurzen Iterationen für dieses Projekt unangebracht erschienen.

2 Autokorrelationsfunktion

Ein zeitunabhängiger Hamiltonian H eines Systems besitzt eine Basis in der er als Multiplikationsoperator wirkt. In dieser Basis besitzt H eine Spektralzerlegung der Form

$$H = \int_{\sigma(H)} \epsilon d\mu(\epsilon)$$

mit Spektralmaß $\mu(\epsilon)$ und Spektrum $\sigma(H)$. Da H nun ein Multiplikationsoperator ist, lässt sich der Zeit-Propagator formal einfach auswerten zu:

$$U(t) = \exp\left(-it \int_{\sigma(H)} \epsilon d\mu(\epsilon)\right) = \int_{\sigma(H)} e^{-it\epsilon} d\mu(\epsilon)$$

Für die Autokorrelationsfunktion $c(t)$ gilt dann dementsprechend:

$$\begin{aligned} c(t) &= \langle \psi, U(t)\psi \rangle = \int_x \psi^* U(t)\psi \\ &= \int_x \psi^* \int_{\sigma(H)} e^{-it\epsilon} d\mu(\epsilon) \psi \\ &= \int_{\sigma(H)} e^{-it\epsilon} \int_x \psi^* d\mu(\epsilon) \psi \\ &= \int_{\sigma(H)} e^{-it\epsilon} d\tilde{\mu}(\epsilon) \end{aligned} \quad (6)$$

Die inverse Fouriertransformation von $c(t)$ gibt ein moduliertes Spektrum der Energieeigenwerte von H .

$$\begin{aligned} c(\epsilon) &= \mathcal{F}[c](\epsilon) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} c(t) e^{it\epsilon} dt \\ &= \frac{1}{\sqrt{2\pi}} \int_{\sigma(H)} \int_{\mathbb{R}} e^{it(\epsilon - \epsilon')} dt d\tilde{\mu}(\epsilon') \\ &= \sqrt{2\pi} \int_{\sigma(H)} \delta(\epsilon - \epsilon') d\tilde{\mu}(\epsilon') \end{aligned} \quad (7)$$

Um die Bedeutung von $c(\epsilon)$ besser zu verstehen, betrachte man einen etwas weniger allgemeinen Fall eines Systems mit isolierten Eigenwerten (H kompakt und normal da selbstadjungiert). In einer Basis in der dieser Hamiltonian diagonal ist hat jeder Zustand die Zerlegung $\psi = \sum_j a_j e_j$ mit Eigenfunktionen e_j zu Eigenenergie ϵ_j . Die Autokorrelation ist nach (6) $c(t) = \sum_j |a_j|^2 e^{-it\epsilon_j}$ und das modulierte Spektrum ist damit nach (7):

$$c(\epsilon) = \sqrt{2\pi} \sum_j |a_j|^2 e^{-it\epsilon_j} \delta(\epsilon - \epsilon') \quad (8)$$

Die δ -Distribution in der letzten Formel kommt natürlich nur dann zustande, wenn über ganz \mathbb{R} integriert wird. Für alle praktischen Zwecke steht aber nur ein begrenztes Intervall $I = [0 : T]$ zur Verfügung. Statt des echten Spektrums bekommt man nach nur die Faltung von $c(\epsilon)$ mit $\mathcal{F}[w](\epsilon)$, wobei $w(t)$ die Rechteck-Fensterfunktion mit Träger in

I ist. Nach Faltungstheorem erhält man:

$$\begin{aligned}
\mathcal{F}[cw](\epsilon) &= \int_{\mathbb{R}} c(t)w(t)e^{it\epsilon} dt \\
&= \int_{\mathbb{R}} \int_{\mathbb{R}} c(t)w(t')\delta(t-t')\frac{2\pi}{(\sqrt{2\pi})^2}e^{it\epsilon} dt dt' \\
&= \frac{1}{(\sqrt{2\pi})^2} \int_{\mathbb{R}} \int_{\mathbb{R}} c(t)w(t') \int_{\sigma(H)} e^{i(t-t')\epsilon'} d\mu(\epsilon') e^{it\epsilon} dt dt' \\
&= \int_{\sigma(H)} \left(\int_{\mathbb{R}} \frac{c(t)}{\sqrt{2\pi}} e^{it\epsilon'} dt \right) \left(\int_{\mathbb{R}} \frac{w(t')}{\sqrt{2\pi}} e^{it'(\epsilon-\epsilon')} dt' \right) d\mu(\epsilon') \\
&= \int_{\sigma(H)} \mathcal{F}[c](\epsilon') \mathcal{F}[w](\epsilon - \epsilon') d\mu(\epsilon') \\
&= (\mathcal{F}[c] * \mathcal{F}[w])(\epsilon) \quad (9)
\end{aligned}$$

Für $w(t) = \Theta(t)\Theta(T-t)$ erhält man

$$W(\epsilon) = \mathcal{F}[w](\epsilon) = \frac{1}{\sqrt{2\pi}} \int_0^T e^{it\epsilon} dt = -i \frac{e^{iT\epsilon} - 1}{\sqrt{2\pi}\epsilon}$$

$$|W(\epsilon)| = \sqrt{\frac{1 - \cos(T\epsilon)}{\pi\epsilon^2}} = \frac{|\sin(\frac{T\epsilon}{2})|}{\pi|\epsilon|}$$

Das Rechteck-Fenster verschmiert also nebeneinander liegende Peaks und verringert die Auflösung des Verfahrens. Man kann das umgehen, indem man eine andere Fenster-Funktion verwendet. Für dieses Projekt wurde die übliche von-Hann Funktion

$$w(t) = \begin{cases} \frac{1 - \cos(\frac{2\pi t}{T})}{2}, & t \in I = [0 : T] \\ 0, & \text{sonst} \end{cases} \quad (10)$$

verwendet.

3 Diskrete Fouriertransformation

Für eine Simulation, die in endlicher Zeit laufen soll, kann man keine direkte Fouriertransformation benutzen, denn die symbolische Auswertung der Integrale ist leider nur für sehr wenige Probleme mit kleinen Gittern sinnvoll implementierbar. Deswegen diskretisiert man bei Simulationen normalerweise den Ort (und Zeit) und geht von kontinuierlichen zu diskreten Fouriertransformationen über.

Sei $x(t)$ ein zunächst kontinuierliches Signal. Durch Festlegen eines Samplingabstands τ erhält man den Dirac-Kamm $s(t) = \delta(t - m\tau)$. Ein diskretisiertes Signal ist dann das punktweise Produkt von $s(t)$ und $x(t)$. Unter der Annahme, dass $x(t)$ periodisch ist mit $x(t) = x(t + n\tau)$ erhält man für die Fouriertransformierte dann:

$$\begin{aligned}
\mathcal{F}[xs](\epsilon) &= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \sum_{m=-\infty}^{\infty} x(t)\delta(t - m\tau)e^{-it\epsilon} dt \\
&= \frac{1}{\sqrt{2\pi}} \sum_{m=-\infty}^{\infty} x(m\tau)e^{-im\tau\epsilon} \\
&= \frac{1}{\sqrt{2\pi}} \sum_{l=-\infty}^{\infty} \left(\sum_{m=0}^{n-1} x(m\tau)e^{-im\tau\epsilon} \right) e^{-iln\tau\epsilon} \\
&= \left(\sum_{m=0}^{n-1} x(m\tau)e^{-im\tau\epsilon} \right) \cdot \frac{1}{\sqrt{2\pi}} \left(\sum_{l=-\infty}^{\infty} e^{-iln\tau\epsilon} \right) \\
&= \frac{1}{N} \tilde{\mathcal{F}}[x](\epsilon) \left(\frac{\sqrt{2\pi}}{\tau} \sum_{l=-\infty}^{\infty} \delta\left(n\epsilon - l\frac{2\pi}{\tau}\right) \right)
\end{aligned}$$

Die Transformierte eines periodischen diskreten Signals ist also selbst von der gleichen Form. Insbesondere folgt aus der letzten Gleichung, dass

$$\epsilon_l = \frac{2\pi}{\tau} \frac{l}{n} = \frac{2\pi}{T} l \quad (11)$$

und die Definition der diskreten Fouriertransformation: Sei $x \in \mathbb{C}^n$ ein Vektor dann ist die DFT von x definiert als

$$\begin{aligned}
\hat{x}_\alpha &= \tilde{\mathcal{F}}[x]_\alpha = N \sum_{m=0}^{n-1} x_m e^{-im\tau\epsilon_\alpha} \\
&= N \sum_{m=0}^{n-1} x_m e^{-im\tau \frac{2\pi}{n\tau} \alpha} = N \sum_{m=0}^{n-1} x_m e^{-i2\pi \frac{m}{n} \alpha} \quad (12)
\end{aligned}$$

Bemerkenswert ist, dass die Gleichung (12) in keinerlei Weise mehr von dem Samplingabstand abhängt, woraus sich die Universalität dieser Transformation ergibt. Die anfangs benutzte Voraussetzung der Periodizität des Signals ist also für jede Funktion auf einem endlichen Intervall erfüllbar: man setzt die Funktion einfach periodisch fort.

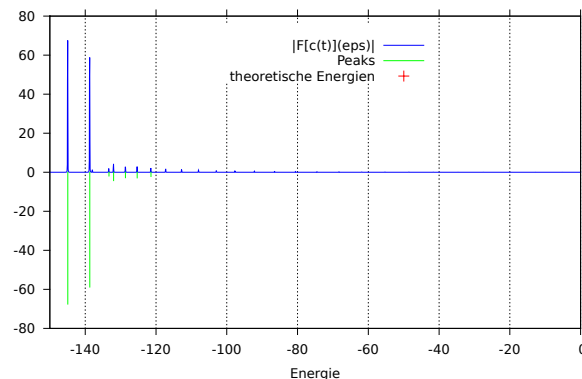
Andererseits muss man bei der Simulation aufpassen, dass der Samplingbereich groß genug gewählt wird, damit die Simulation nie an den Rand des Intervalls kommen kann, was meist unerwünschte Effekte der Periodizität zur Folge hätte.

Die Inverse der DFT berechnet muss bis auf einen Normierungsfaktor die gleiche Gestalt haben:

$$x_k = \tilde{\mathcal{F}}^{-1}[\hat{x}]_k = N \sum_{\alpha=0}^{n-1} \hat{x}_\alpha e^{+i2\pi \frac{\alpha}{n} k} \quad (13)$$

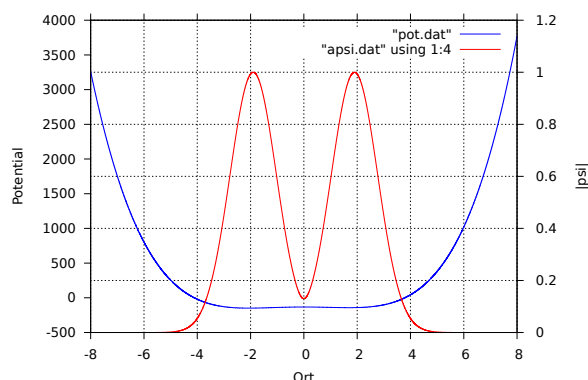
Durch Einsetzen bekommt man dann direkt:

$$\begin{aligned}\tilde{\mathcal{F}}^{-1}[\tilde{\mathcal{F}}[x]]_k &= N^2 \sum_{\alpha=0}^{n-1} \sum_{m=0}^{n-1} x_m e^{-i2\pi \frac{m}{n} \alpha} e^{+i2\pi \frac{\alpha}{n} k} \\ &= N^2 \sum_{m=0}^{n-1} x_m \sum_{\alpha=0}^{n-1} e^{i2\pi \frac{\alpha}{n} (k-m)} \\ &= N^2 \sum_{m=0}^{n-1} x_m n \delta_k^m = N^2 n x_k \Rightarrow N := \frac{1}{\sqrt{n}}\end{aligned}$$



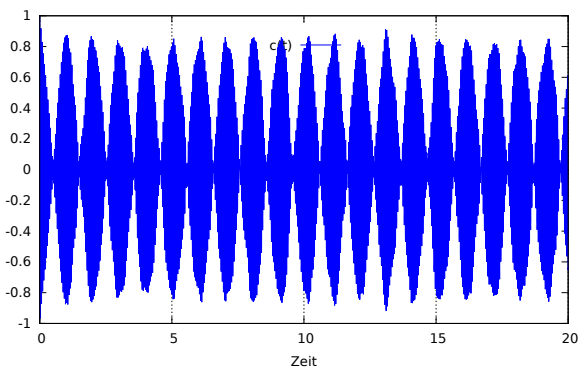
4 Betrachtung einiger Potentiale

4.1 Asymetrischer Doppelpotf



Das erste Potential ist ein einfaches Polynom 4. Ordnung, wobei die Koeffizienten aus (FFS) entnommen wurden. Leider konnte ich die Ergebnisse von (FFS) nicht für die genannten Parameter reproduzieren, die Energien stimmen jedoch für meine Konfiguration mit den gegebenen überein.

Potential	$V(x) = k_0 - k_2 x^2 + k_3 x^3 + k_4 x^4$ $k_0 = -132.7074997, k_2 = 7, k_3 = .5, k_4 = 1$				
Parameter	bins 8192	range [-8 : 8]	steps 10	runs 50k	dt 0.0001
Start-Funktion	$\psi(x) = e^{\frac{(x-a)^2}{2\sigma^2}} + i e^{\frac{(x+a)^2}{2\sigma^2}}$ $a = 1.9, \sigma = 0.87$				



Eigenenergien	
ϵ_α	Amplitude
-144.92	67.53
-138.7	58.83
-133.3	1.91
-131.98	4.28
-128.65	2.8
-125.32	2.89
-121.42	2.18

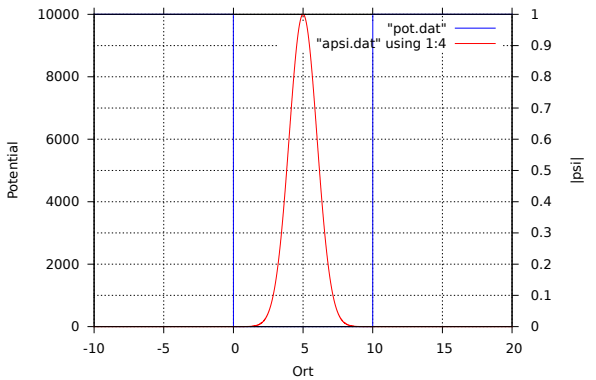
```
local math = require("math")
local complex = require("complex")
local exp = math.exp

config = {
  bins = 4096*4;
  dt = 0.0001;
  range = {-8,8};
  steps = 10; runs = 100000;
  --steps = 10; runs = 100000;
  vstep = 1; vframes = 1;
  --
  potential = function(x)
    --local k0, k2, k3, k4 = -100, 40, 2, 3
    local k0, k2, k3, k4 = -132.7074997, 7, .5, 1
    return k0 - k2*x^2 + k3*x^3 + k4*x^4
  end;
  psi = function(x)
    local a, s = 1.9, 0.87
    return {exp(-(x-a)^2/(2*s^2)), exp(-(x+a)^2/(2*s^2))}
  end;
  energy = function(k)
    local x = k + 1
    return
      -2.5805486322214
      + 9.1857765367527 * x
      + 0.08881127950008 * x^2
      - 0.000260375459505 * x^3
  end;
  enrgrange = {-150, 0, 6};
  output = {
    dir = "./simple";
    apsi = "apsi.dat";
    pot = "pot.dat";
    corr = "corr.dat";
    dftcorr = "dftcorr.dat";
    theoenrg = "theoenrg.dat";
    spectrum = "spectrum.dat";
  };
}
```

../simple.lua

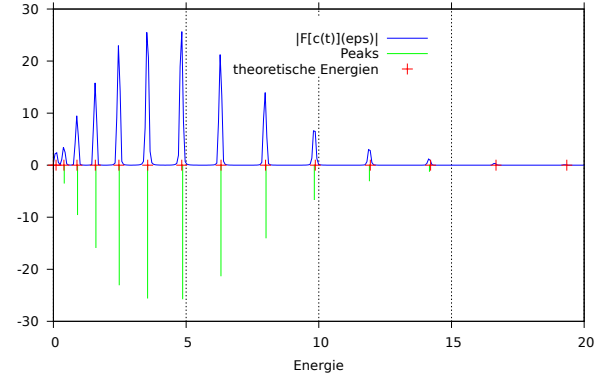
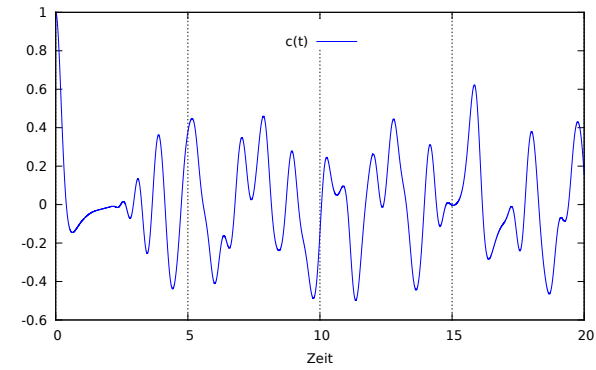
4.2 Potentialkasten

Eigenenergien	
ϵ_α	Amplitude
0.41	3.43
0.91	9.49
1.6	15.81
2.48	22.98
3.55	25.53
4.87	25.66
6.31	21.24
8.01	13.96
9.83	6.59
11.91	2.99
14.17	1.17



Als nächstes versuche ich bekannte Ergebnisse für einen uendlich tiefen Potentialkasten näherungsweise zu reproduzieren.

Potential	$V(x) = a (\Theta(-L/2 - x) + \Theta(-L/2 + x))$ $a = 10000, L = 10$				
Parameter	bins	range	steps	runs	dt
	8192	$[-8 : 8]$	10	$50k$	0.00001
Start-Funktion	$\psi(x) = e^{-\frac{x^2}{2}} e^{i4x}$				



```
local math = require("math")
local complex = require("complex")
local omega = 40;
local L = 10
local exp = math.exp
local sin = math.sin
local sqrt = math.sqrt
local cexp = complex.exp
local pi = math.pi

config = {
  bins = 4096*2; dt = 0.0001;
  range = {-10,20};
  steps = 10; runs = 100000;
  vstep = 100; vframes = 200;
}

potential = function(x)
  if 0 < x and x < L then return 0 else return 10000 end
end;

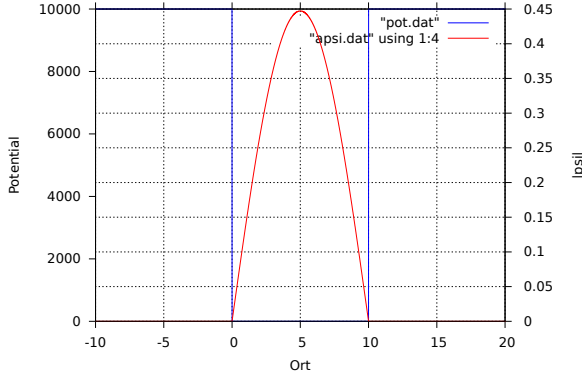
psi = function(x)
  return exp(-(x-L/2)^2/(2)) * cexp({0, 2 * x})
end;

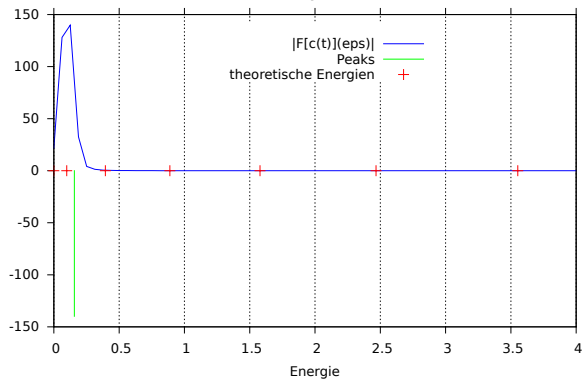
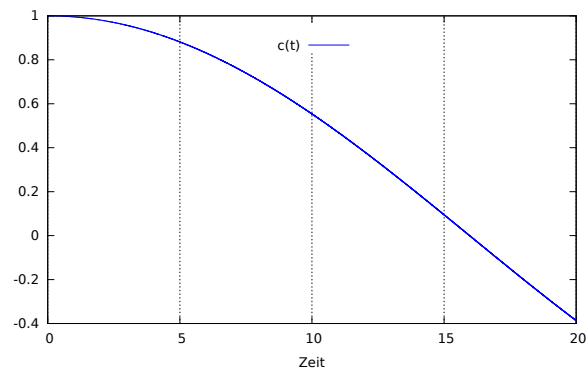
energy = function(k)
  return pi^2 * k^2/L^2
end;

enrgange = {0, 20, 4};
output = {
  dir = "../square";
  apsi = "apsi.dat";
  pot = "pot.dat";
  corr = "corr.dat";
  dftcorr = "dftcorr.dat";
  theoenrg = "theoenrg.dat";
  spectrum = "spectrum.dat";
};
```

../square.lua

4.3 Potentialkasten: Grundzustand





Eigenenergien	
ϵ_α	Amplitude
0.16	139.81

```

local math = require("math")
local complex = require("complex")
local omega = 40;
local L = 10
local exp = math.exp
local sin = math.sin
local sqrt = math.sqrt
local cexp = complex.exp
local pi = math.pi

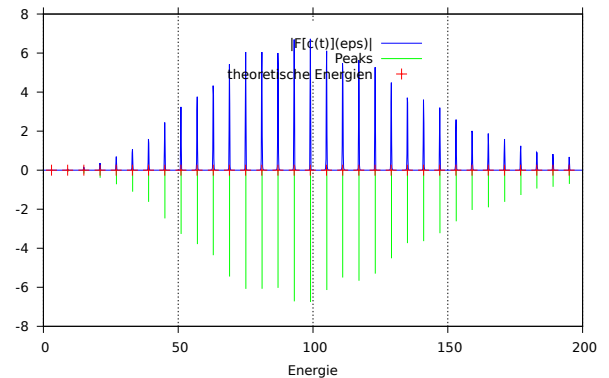
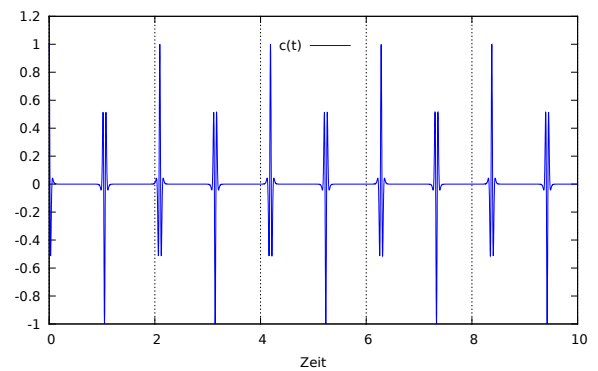
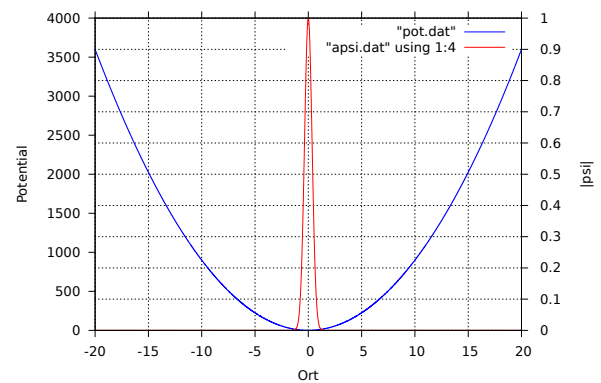
config = {
  bins = 4096*2; dt = 0.0001;
  range = {-10,20};
  steps = 10; runs = 100000;
  vstep = 100; vframes = 200;
}

potential = function(x)
  if 0 < x and x < L then return 0 else return
  10000 end
end;
psi = function(x)
  if 0 < x and x < L then
    return {
      -- + sqrt(2/L) * sin(2*pi/L * x)
      sqrt(2/L) * sin(1*pi/L * x),
      0
    }
  else
    return {0,0}
  end
end;
energy = function(k)
  return pi^2 * k^2/L^2
end;
enrgrange = {0, 4, 6, 3};
output = {
  dir = "../square1.lua";
  apsi = "apsi.dat";
  pot = "pot.dat";
  corr = "corr.dat";
  dftcorr = "dftcorr.dat";
  theoeng = "theoeng.dat";
  spectrum = "spectrum.dat";
};

```

../square1.lua

4.4 Harmonischer Oszillator



Eigenenergien	
ϵ_α	Amplitude
9.11	$5.09 \cdot 10^{-2}$
15.02	0.15
21.05	0.37
27.08	0.7
33.11	1.08
39.02	1.6
45.05	2.44
51.08	3.24
57.11	3.76
63.02	4.33
69.05	5.42
75.08	6.06
81.12	6.05
87.02	6.01
93.05	6.69
99.09	6.73
105.12	6.12
111.02	5.47
117.06	5.64
123.09	5.27
129.12	4.49
135.03	3.71
141.06	3.61
147.09	3.2
153.12	2.59
159.03	2.01
165.06	1.88
171.09	1.6
177.12	1.24
183.03	0.92
189.06	0.82
195.09	0.68
201.12	0.51
207.03	0.36
213.06	0.32
219.09	0.25
225.13	0.19
231.16	0.13
237.06	0.11
243.1	$8.47 \cdot 10^{-2}$
249.13	$6.11 \cdot 10^{-2}$
255.16	$4.04 \cdot 10^{-2}$
261.07	$3.33 \cdot 10^{-2}$

```

local math = require("math")
local complex = require("complex")
local omega = 6;
local exp = math.exp
local sqrt = math.sqrt

config = {
  bins = 4096*2; dt = 0.0001;
  range = {-20,20};
  steps = 10; runs = 50000;
  vstep = 100; vframes = 200;
}

potential = function(x)
  return omega^2/4 * x ^ 2;
end;
psi = function(x)
  local a = .5
  local aa = a * a
  local x0 = 0
  local k0 = 10
  local xx = (x-x0) * (x-x0)
  return math.exp(-xx/(aa)) * complex.exp({0, k0
    *(x)})
end;
energy = function(k)
  return omega * (.5 + k)
end;

```

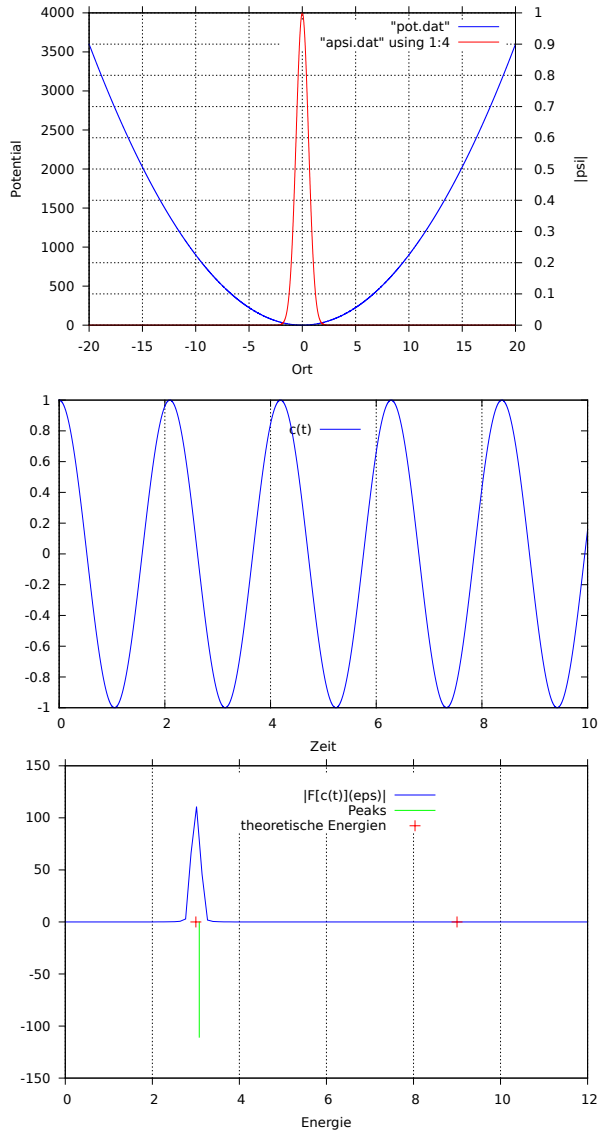
```

enrgange = {0, 200, 6, 1.1};
output = {
  dir = "/harmosca";
  apsi = "apsi.dat";
  pot = "pot.dat";
  corr = "corr.dat";
  dftcorr = "dftcorr.dat";
  theoenrg = "theoenrg.dat";
  spectrum = "spectrum.dat";
};
}

../harmosca.lua

```

4.5 Harmonischer Oszillator: Grundzustand



Eigenenergien	
ϵ_α	Amplitude
3.08	110.71

```

local math = require("math")
local complex = require("complex")
local omega = 6;
local exp = math.exp
local sqrt = math.sqrt

config = {
  bins = 4096*2; dt = 0.0001;

```

```

range = {-20,20};
steps = 10; runs = 50000;
vstep = 100; vframes = 200;

potential = function(x)
    return omega^2/4 * x ^ 2;
end;
psi = function(x)
    return {
        exp(-omega/4*x^2),
        0
    }
end;
energy = function(k)
    return omega * (.5 + k)
end;
enrgrange = {0, 2*omega, 6, 3.5};
output = {
    dir = "../harmosc1";
    apsi = "apsi.dat";
    pot = "pot.dat";
    corr = "corr.dat";
    dftcorr = "dftcorr.dat";
    theoenrg = "theoenrg.dat";
    spectrum = "spectrum.dat";
};
}

../harmosc1.lua

```

Quellen

- [FFS] M.D Feit, J.A Fleck Jr., A Steiger *Solution of the Schrödinger equation by a spectral method* [http://dx.doi.org/10.1016/0021-9991\(82\)90091-2](http://dx.doi.org/10.1016/0021-9991(82)90091-2)
- [FMF] J. A. Fleck, J. R. Morris and M. D. Feit *Time-dependent propagation of high energy laser beams through the atmosphere* <http://dx.doi.org/10.1007/BF00896333>
- [MA] Mrinal Mandal, Amir Asif *Continuous and Discrete Time Signals and Systems* Kap.12.1 http://www.cambridge.org/resources/0521854555/4421_Chapter%2012%20-%20Discrete%20Fourier%20transform.pdf