

# 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  $\tau$  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:

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

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

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

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  $\epsilon_j$ . 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  $\delta$ -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  $\tau$  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:

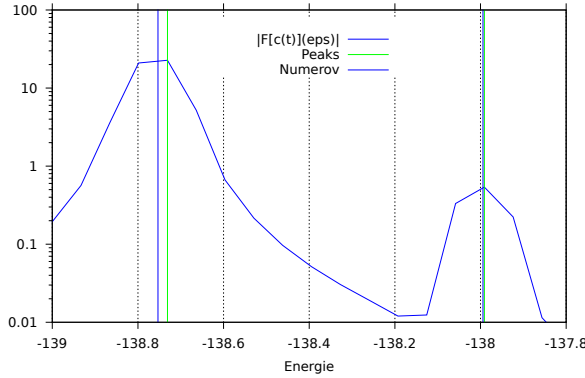
$$x_k = \tilde{\mathcal{F}}^{-1}[\hat{x}]_k = N \sum_{\alpha=0}^{n-1} \hat{x}_\alpha e^{+i2\pi \frac{\alpha}{n} k} \quad (13)$$

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 Suchalgorithmus und sein Grenzen



Peaksuche	Numerov
-138.7310695	-138.7531872
-137.9909911	-137.9943586

Für die Auswertung der Potentiale wähle ich einen Suchalgorithmus der sich leider prinzipbedingt als sehr fehleranfällig herausstellen sollte.

Der eigentliche Suchalgorithmus stammt von (SAPS) und benutzt die einfachste Signifikanzfunktion  $S_1$ . Da dies meistens nicht ausreicht um alle Peaks zu finden werden nach einer Iteration alle gefundenen Peaks innerhalb des zugehörigen Fensters mithilfe der Funktion  $f(x) = e^{w/|x-x_{\text{peak}}|}$  „gelöscht“, worauf eine erneute Iteration folgt, bei der allerdings Suchschranke (Forderung in Standardabweichungen) verdoppelt wird, um eine Konvergenz des Verfahrens zu erzwingen.

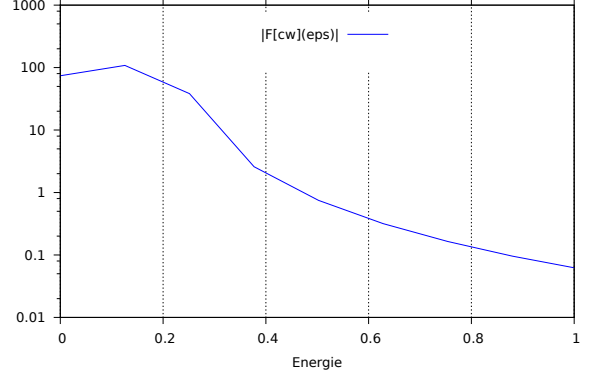
**Start** • berechne Werte  $a_i = S(d_i)$ , alle Werte  $> 0$  sind potentielle Peaks

**Iteration** • vergrößere peak Fenster  $w$  um 1.5

- verdopple Suchschranke  $h$
- berechne Mittelwert  $m$  von  $\{a_i\}$
- berechne Standardabweichung  $s$  von  $\{a_i\}$
- alle Werte  $> 0$ , die um das  $h$ -Fache von  $s$  abweichen im Fenster eines Peaks
- verkleinere jedes Fenster solange, bis nur der maximale Wert bleibt; dieser Wert ist der gesuchte Peak

- breche ab, falls keine Peaks gefunden wurden; andernfalls lösche alle Peaks aus  $\{a_i\}$  und beginne nächste Iteration

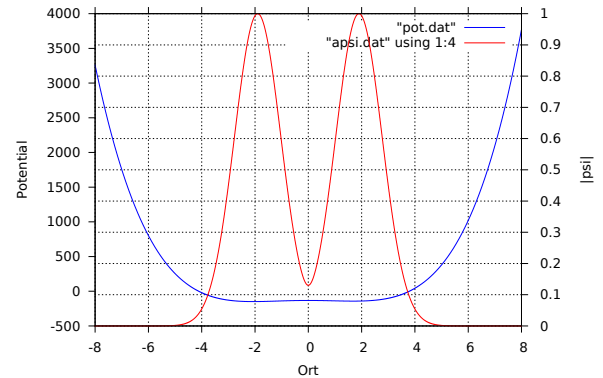
Wie man aus der oberen Abbildung erkennen kann entsprechen sind die gefundenen Peaks nicht immer den tatsächlichen Energiewerten, weil durch die Diskretisierung sich das Maximum deutlich vom richtigen Wert verschieben kann.



Leider schlägt der Algorithmus manchmal sogar komplett fehl, wie hier beim Grundzustand des im weiteren betrachteten Kastenpotentials. Der Peaks befindet sich am Rand und unterscheidet sich nicht all zu sehr von seiner Umgebung, was dazu führt, dass der Suchalgorithmus ihn ignoriert.

Wie schon in der oberen Tabelle werde ich im Weiteren die gefundenen Energiewerte mit der Numerov-Approximation für die stationäre Schrödingergleichung vergleichen. (Das hat den Nachteil, dass man die Potentiale nur relativ kleine Werte annehmen müssen, damit die Approximation noch innerhalb der doppelten Genauigkeit berechnet werden kann.)

### 4.2 Asymmetrischer Doppeltopf



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 mehr oder

weniger überein. (Die Parameter aus (FFS) sind offensichtlich falsch ein  $\tau = 5.7$  entspricht einer maximalen simulierbaren Energie von  $\approx 0.551157$ , die behaupteten Energien von  $-144$  können also niemals aus einer Simulation mit diesen Parametern entstanden sein. Man erhält wiederum ähnliche Werte wenn man sich um 4(!) Größenordnungen von den angegebenen Parametern entfernt, d.h.  $\tau = \frac{5.7}{1000}$  und  $\xi = \frac{0.825}{10}$ ).

Die Parameter werden an die Simulation mit Hilfe eines einfachen Lua-Skripts übergeben, in dem die Tabelle config definiert wird.

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

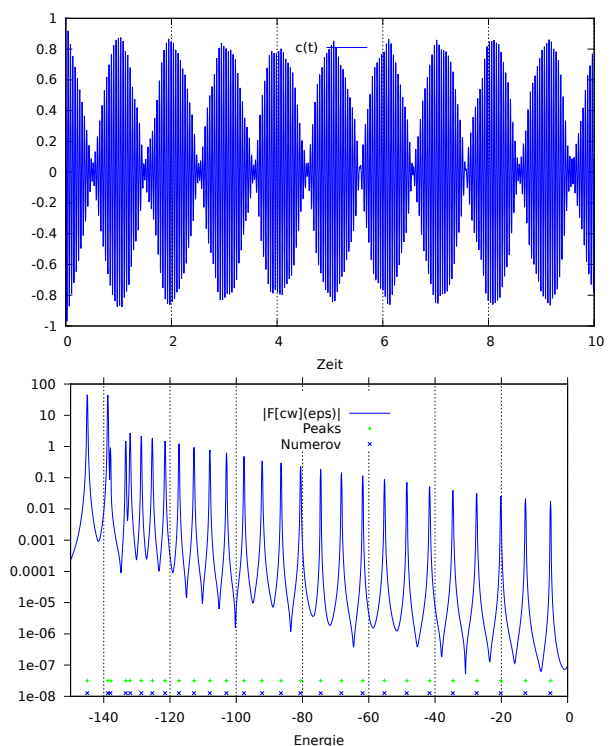
config = {
  bins = 4096;
  dt = 0.0001;
  range = {-8,8};
  steps = 10; runs = 50000;
  vstep = 1; vframes = 1;

  potential = function(x)
    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;
  enrgrange = {-150, 0, 6, 1.5};
  output = {
    dir = "./simple";
    apsi = "apsi.dat";
    pot = "pot.dat";
    corr = "corr.dat";
    dftcorr = "dftcorr.dat";
    spectrum = "spectrum.dat";
  };
}
```

../simple.lua

Die einzelnen Variablen sind (Grau = optional)

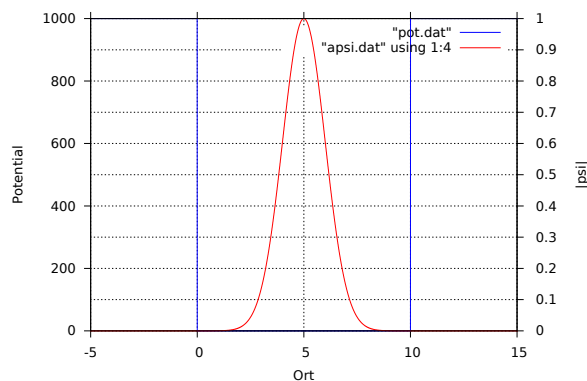
bins	Anzahl der Stützstellen im Ort
dt	einzelner Zeitschritt ( $\tau$ )
range	das betrachtete Ortsintervall $[-8, 8]$
steps	$\tau$ zwischen Korrelationsmessungen
runs	Anzahl Korrelationsmessungen
vstep	Zeitschritte zwischen Frames
vframes	Anzahl Frames
potential(x)	reelles Potential ( $V(x)$ )
psi(x)=z	Wellefunktion $z=\{x,y\}$ ( $\psi(x)$ )
energie(k)	exakte Energien $\epsilon_k$
enrgrange	betrachteten Energien Min.,Max.,Suchfenster, Suchschranke
output	
.dir	Ausgabeverzeichnis
.apsi	Werte von $\psi(x, t = 0)$
.pot	Werte von $V(x)$
.corr	Werte von $c(t)$
.dftcorr	Werte von $\mathcal{F}[cw](\epsilon)$
.spectrum	gefundene Peaks



Peaksuche	Numerov
-145.0159169	-144.9662986
-138.7327316	-138.7531872
-137.8530856	-137.9943586
-133.3291922	-133.3544215
-132.0725552	-132.017109
-128.6796351	-128.6551459
-125.286715	-125.3398237
-121.5168038	-121.4735992
-117.2442378	-117.2776404
-112.7203444	-112.7713057
-107.9451236	-107.9886065
-102.9185753	-102.9525932
-97.6406967	-97.68213261
-92.23716031	-92.19280618
-86.45662983	-86.49781338
-80.55043564	-80.60852286
-74.51857774	-74.53485882
-68.23539244	-68.28557958
-61.82654342	-61.86848533
-55.2920307	-55.29057713
-48.50619057	-48.55818105
-41.59468673	-41.67704696
-34.55751919	-34.6524282
-27.39468794	-27.48914674
-20.10619298	-20.19164709
-12.69203432	-12.76404124
-5.152211952	-5.210146622

Wie schon eingangs erwähnt, leidet die benutzte Suchmethode an der Diskretisierung, sodass die Werte im Rahmen von  $2\pi/\tau n$  schwanken.

### 4.3 Potentialkasten



Als nächstes versuche ich bekannte Ergebnisse für einen unendlich tiefen Potentialkasten näherungsweise zu reproduzieren. Um vergleichswerte der Numerov-Approximation zu erhalten, musste ich das Potential auf 1000 reduzieren, da ich sonst an die Grenzen der Genauigkeit des double Datentyps stoßen würde.

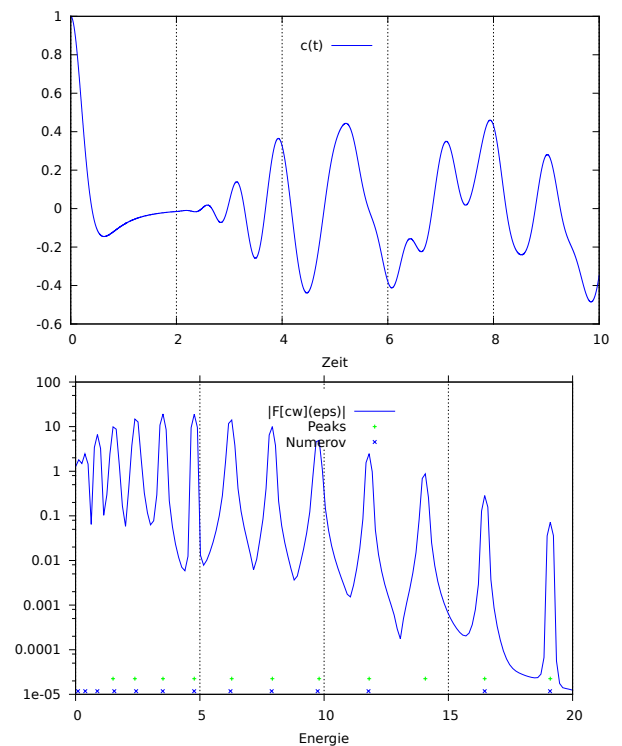
```
local math = require("math")
```

```
local complex = require("complex")
local L = 10
local exp = math.exp
local cexp = complex.exp
local pi = math.pi

config = {
  bins = 4096; dt = 0.0001;
  range = {-5,15};
  steps = 10; runs = 50000;
  vstep = 100; vframes = 200;
}

potential = function(x)
  if 0 < x and x < L then return 0 else
    return 1000 end
end;
psi = function(x)
  return exp(-(x-L/2)^2/(2)) * cexp({0,
    2 * x})
end;
energy = function(r)
  local k = r + 1
  return pi^2 * k^2/L^2
end;
enrgrange = {0, 20, 8, 2};
output = {
  dir = "../square";
  apsi = "apsi.dat";
  pot = "pot.dat";
  corr = "corr.dat";
  dftcorr = "dftcorr.dat";
  spectrum = "spectrum.dat";
};
}
```

../square.lua



Peaksuche	Numerov	exakt
	0.0974096378	0.09869604401
	0.3896383141	0.394784176
	0.8766853172	0.8882643961
1.507964474	1.55854946	1.579136704
2.387610417	2.435229078	2.4674011
3.518583772	3.506722029	3.553057584
4.775220833	4.773025686	4.836106157
6.283185307	6.23413694	6.316546817
7.916813487	7.890052188	7.994379565
9.801769079	9.740767336	9.869604401
11.81238838	11.78627779	11.94222133
14.07433509		14.21223034
16.4619455	16.46166369	16.67963144
19.10088333	19.09152739	19.34442463

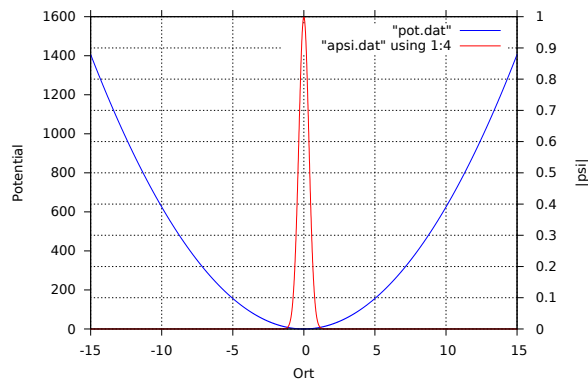
```

dir = "./harmosca";
apsi = "apsi.dat";
pot = "pot.dat";
corr = "corr.dat";
dftcorr = "dftcorr.dat";
spectrum = "spectrum.dat";
};
}

```

../harmosca.lua

#### 4.4 Harmonischer Oszillator



```

local math = require("math")
local complex = require("complex")
local omega = 5;
local exp = math.exp
local sqrt = math.sqrt
local pi = math.pi

config = {
  bins = 4096; dt = 1 / (1000 * pi);
  range = {-15,15};
  steps = 10; runs = 50000;
  vstep = 100; vframes = 200;
}

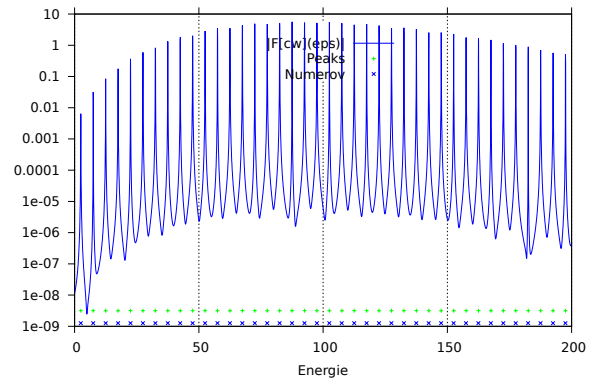
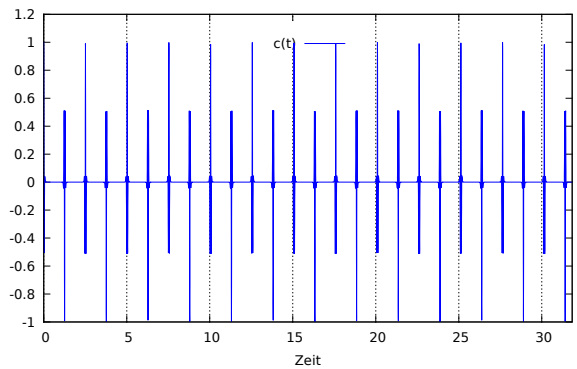
potential = function(x)
  return omega^2/4 * x ^ 2;
end;

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;

enrgrange = {0, 200, 6, 1};
output = {

```



Peaksuche	Numerov	exakt
2.487140309	2.5	2.5
7.500899345	7.499999998	7.5
12.51465838	12.49999999	12.5
17.488939	17.49999998	17.5
22.50269803	22.49999995	22.5
27.51645707	27.49999992	27.5
32.49073769	32.49999987	32.5
37.50449672	37.4999998	37.5
42.51825576	42.49999971	42.5
47.5320148	47.49999959	47.5
52.50629541	52.49999945	52.5
57.52005445	57.49999928	57.5
62.53381349	62.49999908	62.5
67.5080941	67.49999884	67.5
72.52185314	72.49999856	72.5
77.53561217	77.49999824	77.5
82.50989279	82.49999788	82.5
87.52365183	87.49999748	87.5
92.53741086	92.49999702	92.5
97.51169148	97.49999651	97.5
102.5254505	102.4999959	102.5
107.5392096	107.4999953	107.5
112.5134902	112.4999946	112.5
117.5272492	117.4999939	117.5
122.5410082	122.4999931	122.5
127.5152889	127.4999922	127.5
132.5290479	132.4999913	132.5
137.5428069	137.4999902	137.5
142.556566	142.4999891	142.5
147.5308466	147.4999879	147.5
152.5446056	152.4999867	152.5
157.5583647	157.4999853	157.5
162.5326453	162.4999839	162.5
167.5464043	167.4999823	167.5
172.5601633	172.4999807	172.5
177.534444	177.499979	177.5
182.548203	182.4999772	182.5
187.561962	187.4999752	187.5
192.5362427	192.4999732	192.5
197.5500017	197.4999711	197.5

## 5 Fazit

Wie man aus der betrachtung der Spektren entnehmen kann, erlaubt die Autokorrelations-Methode ziemlich zuverlässige und gut reproduzierbare Spektren zu generieren, d.h. die Spektren verändern sich nicht all zu sehr bei kleinen Veränderungen der Parameter der Simulation. Um die eigentlichen Energiewerte zu bestimmen, müsste man die gewonnenen Peaks geeignet Fitten.

## 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](http://www.cambridge.org/resources/0521854555/4421_Chapter%2012%20-%20Discrete%20Fourier%20transform.pdf)
- [SAPS] Girish Keshav Palshikar *Simple Algorithms for Peak Detection in Time-Series* [http://www.tcs-trddc.com/trddc\\_website/pdf/SRL/Palshikar\\_SAPDTS\\_2009.pdf](http://www.tcs-trddc.com/trddc_website/pdf/SRL/Palshikar_SAPDTS_2009.pdf)