
Computerphysik

Übungsblatt 7

SS 2013

Website: <http://www.thp.uni-koeln.de/trebst/Lectures/2013-CompPhys.html>

Abgabedatum: Montag, 10. Juni 2013 vor Beginn der Vorlesung

Auf diesem Übungsblatt wollen wir uns noch einmal mit der numerischen Lösung von Differentialgleichungen beschäftigen und dabei eine Reihe von Problemstellungen aus dem Bereich der **Quantenmechanik** betrachten. Davor wie immer eine kurze Aufgabe zur Programmieretechnik.

25. Als die Funktionen laufen lernten

Programmiertechniken

Nachdem wir schon diverse Animationen mit Python erzeugt haben, wollen wir uns heute dem Animieren von Funktionen widmen. Das Vorgehen ist praktisch identisch wie bisher: Wir erneuern die Daten des Plots in jedem Schritt und lassen diesen neu darstellen. Als einfaches Beispiel wollen wir die Betragsquadrate zweier eindimensionaler Gauß'schen Wellenpakete plotten, die sich entlang der x-Achse fortbewegen. Dazu der folgende Code [animation_wave.py](#):

```
import numpy as np
import matplotlib.pyplot as plt

# Ein freies Gaussches Wellenpaket zur Zeit t mit Impuls p0
# und Breite sigma:
def gaussian_wave(x, p0, t, sigma):
    return ((sigma/np.sqrt(2.0*np.pi))**(0.5)*
            (sigma**2 + 1j*t*0.5)**(-0.5)*
            np.exp(-sigma**2*p0**2)*
            np.exp(-(x-2j*sigma**2*p0)**2/(4*(sigma**2+0.5j*t))))

t, dt = 0., 0.1
# x enthaelt 1000 Werte zwischen -10 und 100:
x = np.linspace(-10, 100, 1000)

# wir plotten erstmal "leere" Funktionen
line1, = plt.plot([], [])
line1.set_xdata(x)
line2, = plt.plot([], [])
line2.set_xdata(x)

# x und y Begrenzungen des Plots:
plt.xlim(0, 100)
plt.ylim(-0.1, 0.5)

# Jetzt folgt die (Endlos-)Animationsschleife:
while True:
    t += dt
    # Welle 1 mit sigma = 1
    wave1 = abs(gaussian_wave(x, 2.0, t, 1.))**2
    line1.set_ydata(wave1)
    # Welle 2 mit sigma = 2
    wave2 = abs(gaussian_wave(x, 2.0, t, 2.0))**2
    line2.set_ydata(wave2)
    plt.pause(0.01)
plt.show()
```

Die `pause()`-Funktion von PyPlot kennen Sie ja schon. Neu ist die Verwendung von Linien-Objekten. Der Befehl `plot()` gibt eine Liste von eben diesen Objekten zurück. Die etwas gewöhnungsbedürftige Syntax `line0, = plt.plot([], [])` speichert dieses (in unserem Fall erste und einzige Objekt der zurückgegebenen Liste) in der Variablen `line0`. In der Animationsschleife müssen wir nun diesem Objekt die Werte für die X- und Y-Achsen zuweisen. Das geschieht in `set_xdata()`, welches wir nur einmal aufrufen müssen, denn die x-Achse ändert sich während der Simulation nicht, und dem wiederholten Aufruf von `set_ydata()`, was zur eigentlichen Animation führt.

In der Animation haben die beiden Wellenpakete eine unterschiedliche Anfangsbreite. Mit der Zeit “zerfließen” die Pakete, d.h. sie werden breiter im Ortsraum. Wieso zerfließt die zu Beginn weniger breite Welle schneller?

26. Allerlei aus dem Potentialtopf

5 Punkte

In unserem ersten Beispiel wollen wir ein quantenmechanisches Teilchen verfolgen, welches wir nicht nur in eine Raumdimension zwingen sondern auch noch in einen sogenannten **Potentialtopf** sperren. Eine derartige Situation lässt sich dann durch die *zeitunabhängige* Schrödinger-Gleichung in einer Dimension

$$\left[-\frac{1}{2} \frac{d^2}{dx^2} + V(x) \right] \psi(x) = E \psi(x), \quad (1)$$

beschreiben, wobei wir $\hbar = m = 1$ gesetzt haben. Beachten Sie, dass es sich hierbei um eine *gewöhnliche* Differentialgleichung handelt.

Diese Schrödinger-Gleichung wollen wir nun für verschiedene Potentiale $V(x)$ — etwa jene, die in Abbildung 1 dargestellt sind — anhand des in der Vorlesung vorgestellten **Numerov-Algorithmus** numerisch exakt lösen.

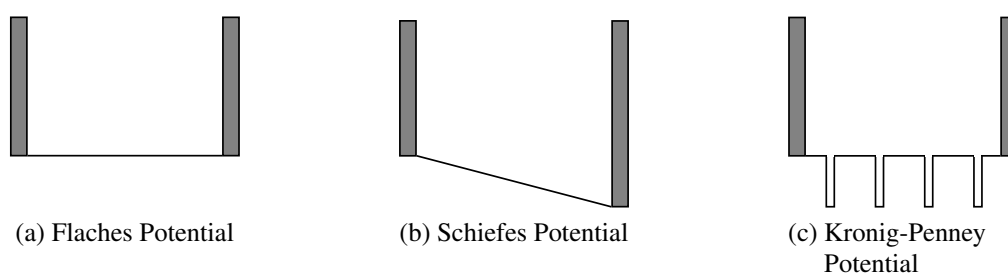


Abbildung 1: Die drei verschiedenen Potentialtöpfe.

Dazu können wir wie folgt vorgehen: Wir diskretisieren die x -Achse im Bereich $-1.0 \leq x \leq 1.0$ in 1000 Abschnitte der Größe $\Delta x = 0.002$. Des weiteren benötigen wir für das Numerov-Verfahren, in welchem wir die Schrödinger-Gleichung etwa vom linken zum rechten Rand entlang der obigen Diskretisierungsschritte integrieren wollen, zwei Startwerte. Dazu vergegenwärtigen wir uns zum einen, dass die Wellenfunktion am (linken) Rand des Potentialtopfes wegen der unendlich hohen Wände verschwinden muss, wohingegen die Wellenfunktion innerhalb des Potentialtopfes durchaus einen endlichen Wert annehmen kann. Dies wollen wir mit

den folgenden Startwerten des Numerov-Verfahrens in der Nähe des linken Randes wie folgt modellieren:

$$\begin{aligned}\psi(x = -1.000) &= 0, \\ \psi(x = -0.998) &= 0.5,\end{aligned}$$

wobei die exakte Wahl des endlichen Wertes für $\psi(x = -0.998)$ unerheblich ist, da wir die Wellenfunktion nach Integration durch das Numerov-Verfahren unabhängig von dieser Wahl normieren müssen.

Führen Sie die Numerov-Integration für die angegebenen Potentiale und verschiedene fixe Energien E aus, etwa für 250 äquidistante Energiewerte im Bereich $-5 \leq E \leq 95$ und plotten Sie jeweils den Wert von $\psi(x = +1)$ als Funktion der Energie.

Die Nullstellen dieser Funktion entsprechen die erlaubten Eigenenergien, denn nur dann wird auch die Randbedingung am *rechten* Rand erfüllt, wo die Wellenfunktion ebenfalls verschwinden muss, d.h. $\psi(x = +1) = 0$.

Nun wollen wir durch Intervallschachtelung/Bisektion um die kleinste dieser Nullstellen die Energie des Grundzustands bis zu einer Genauigkeit von 10^{-9} berechnen. (Die Intervallschachtelung wird also so lange durchgeführt, bis $|\psi(1)| < 10^{-9}$ gilt.) Plotten Sie außerdem die Grundzustandswellenfunktion und das jeweilige Potential in einen Plot.

Die zu bearbeitenden **Potentiale** sind in Abb. 1 schematisch dargestellt, wobei wir für das in (c) abgebildete Kronig-Penney Potential zwei verschiedene Rechnungen machen wollen. In allen Fällen approximieren wir die “unendlich hohen” Wände mittels

$$V(x) = 10^9 \quad \text{für } |x| \geq 1.$$

Der innere Bereich, $|x| < 1$, sei jeweils gegeben via

- (a) $V(x) = 0$,
- (b) $V(x) = -10x$,
- (c) Das Kronig-Penney Modell, welches die periodischen Potentialtöpfe in einem kristallinen Festkörper modellieren soll.

$$V(x) = \begin{cases} -V_0 & -0.78 < x < -0.72 \\ -V_0 & -0.28 < x < -0.22 \\ -V_0 & 0.22 < x < 0.28 \\ -V_0 & 0.73 < x < 0.78 \\ 0 & \text{sonst.} \end{cases}$$

Berechnen sie dabei den Grundzustand einmal für $V_0 = 2$ und einmal für $V_0 = 1000$.

Zusatzfrage: Sie wissen, dass wir in der Quantenmechanik mit komplexen Zahlen rechnen müssen. Wieso können wir bei dieser Aufgabe komplett innerhalb der reellen Zahlen bleiben?

27. Pendeln in der Quantenwelt

5 Punkte

In unserem zweiten Beispiel wollen wir den quantenmechanischen **harmonischen Oszillator** betrachten, dessen Potential $V(x) = \omega^2 x^2 / (2m)$ mit der Frequenz ω und der Teilchenmasse m gegeben sei. Wir setzen wieder $\hbar = \omega = m = 1$. Der harmonischen Oszillator wird also im allgemeinen durch eine *zeitabhängige* Schrödinger-Gleichung der Form

$$\frac{1}{2} \left[-\frac{\partial^2}{\partial x^2} + x^2 \right] \psi(x, t) = i \frac{\partial}{\partial t} \psi(x, t) \quad (2)$$

beschrieben.

Mit dem Produktansatz $\psi(x, t) = \phi(x)\chi(t)$, d.h. der Separation von Orts- und Zeitkoordinate, reduziert sich die Schrödinger-Gleichung auf *zwei gewöhnliche* Differentialgleichungen. Die Lösungen für den Ortsanteil lassen sich über die sogenannten **Hermite-Polynome** H_n darstellen:

$$\phi_n(x) = \pi^{-\frac{1}{4}} \frac{1}{\sqrt{2^n n!}} H_n(x) e^{-\frac{x^2}{2}}, \quad \text{mit } n = 0, 1, 2, \dots$$

Die Hermite-Polynome $H_n(x)$ können mit folgender Vorschrift rekursiv konstruiert werden:

$$\begin{aligned} H_0(x) &= 1, \\ H_1(x) &= 2x, \\ H_{n+1}(x) &= 2xH_n(x) - 2nH_{n-1}(x). \end{aligned}$$

Die Lösung für den Zeitanteil der Schrödinger-Gleichung (2) hat die Form

$$\chi_n(t) = e^{-iE_n t},$$

wobei die Energie des n -ten Zustands gegeben ist als $E_n = n + \frac{1}{2}$.

Beachten Sie, dass die Funktionen $\psi_n(x, t) = \phi_n(x)\chi_n(t)$ lediglich eine Basis des Lösungsraums von Gleichung (2) bilden, d.h. wir können eine allgemeine Lösung als Linearkombination dieser Basiselemente schreiben als

$$\psi(x, t) = \sum_{n=0}^{\infty} c_n \phi_n(x) \chi_n(t). \quad (3)$$

Wir wollen nun für eine beliebige vorgegebene Wellenfunktion die Koeffizienten c_n berechnen. Dafür machen wir uns die Tatsache, dass die Funktionen ψ_n eine Orthogonalbasis darstellen, zunutze. Das Skalarprodukt $\langle \cdot | \cdot \rangle$ auf dem Raum der Lösungen ist für zwei beliebige Wellenfunktionen ψ_n und ψ_m gegeben als

$$\langle \psi_n(x, t) | \psi_m(x, t) \rangle = \int_{-\infty}^{\infty} dx \psi_n^*(x, t) \psi_m(x, t).$$

Bekanntermaßen gilt für das Skalarprodukt zweier orthonormalen Basisvektoren immer

$$\langle \psi_n(x, t) | \psi_m(x, t) \rangle = \delta_{n,m},$$

d.h. wir können die Koeffizienten c_n des allgemeinen Zustands (siehe Gl. (3)) $\psi(x, t)$ durch

Projektion des Zustands auf die Basisvektoren finden:

$$c_n = \langle \psi_n(x, t) | \psi(x, t) \rangle.$$

Gegeben seien nun folgende Wellenfunktionen zum Zeitpunkt $t = 0$:

$$\psi_A(x, t = 0) = \pi^{-1/4} e^{-\frac{1}{2}(x-2)^2}, \quad (4)$$

$$\psi_B(x, t = 0) = \frac{1}{\pi} \frac{0.05}{(x-2)^2 + 0.05^2}. \quad (5)$$

Berechnen sie numerisch die **Zeitentwicklung** dieser beiden Wellenfunktionen und plotten Sie etwa das Betragsquadrat $|\psi_{A/B}(x, t)|^2$ als eine Animation in der Zeit. Bestimmen Sie dazu zunächst die Koeffizienten c_n , wobei wir uns hier auf die ersten zehn beschränken wollen. Das Integral im Skalarprodukt können Sie z.B. mit der in der Vorlesung beschriebenen Trapezmethode berechnen. Die Zeitentwicklung folgt dann direkt aus Gleichung (3). Wir schlagen vor, die Funktionen im Intervall $x \in [-6.0, 6.0]$ für etwa 1000 Stützstellen zu berechnen. Als Zeitschritt können Sie etwa $\Delta t = 0.05$ wählen.

28. Zeitaufgelöste Streuung an Potentialbarrieren

optionale Aufgabe – 12 Punkte

In dieser Bonusaufgabe wollen wir die zeitaufgelöste Streuung eines auf eine Potentialbarriere zulaufenden Wellenpakets simulieren (siehe Abbildung 2) und damit einige fundamentale **quantenmechanische Phänomene illustrieren**.

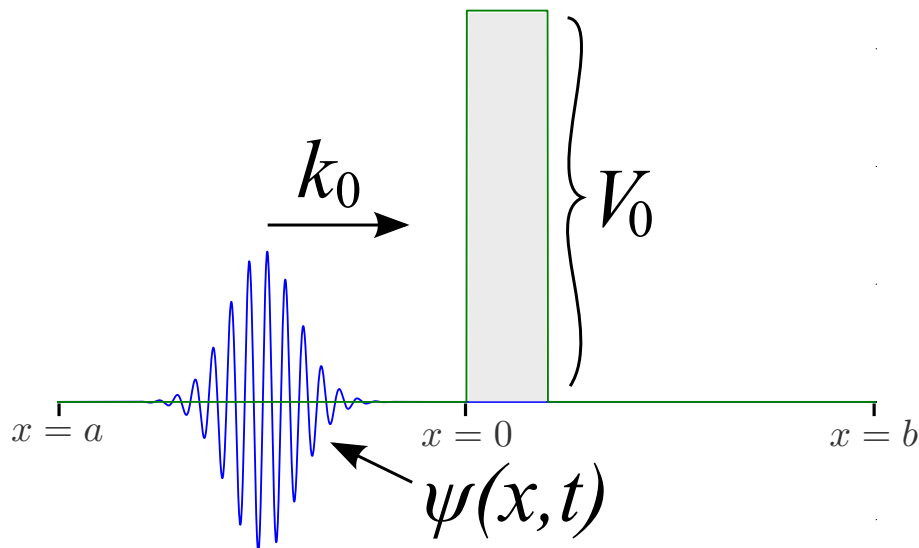


Abbildung 2: Schematische Darstellung eines Wellenpakets, welches auf eine Potentialbarriere zuläuft.

Bevor wir uns in die Simulation stürzen können, werden wir Sie noch mit dem einen oder anderen theoretischen Kniff vertraut machen, der eine derartige Simulation erheblich vereinfacht.

Theorie

Wir starten dazu mit der zeitabhängigen Schrödingergleichung für ein Teilchen in einer Raumdimension:

$$\left[-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V(x) \right] \psi(x,t) = i\hbar \frac{\partial}{\partial t} \psi(x,t). \quad (6)$$

Nun wollen wir eine clevere Methode zur Lösung verwenden, die auf die **Fourier-Transformation** zurückgreift. Wir verwenden folgende Konvention für Hin- und Rücktransformationen:

$$\tilde{\psi}(k,t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dx \psi(x,t) e^{-ikx}, \quad \psi(x,t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dk \tilde{\psi}(k,t) e^{ikx}.$$

Setzen wir die Fourier-Transformation von $\psi(x,t)$ in Gl. (6) ein, so finden wir:

$$\left[\frac{\hbar^2 k^2}{2m} + V\left(i\frac{\partial}{\partial k}\right) \right] \tilde{\psi}(k,t) = i\hbar \frac{\partial}{\partial t} \tilde{\psi}(k,t). \quad (7)$$

Lassen Sie sich von der ungewohnten Notation $V(i\partial/\partial k)$ nicht abschrecken: Wir werden sie im folgenden nicht weiter berücksichtigen müssen. Allerdings entdecken wir eine interessante Symmetrie der Gleichungen (6) und (7): In beiden Fällen besteht die linke Seite aus einer Multiplikation der Wellenfunktion mit einem einfachen Term und einem scheinbar etwas komplizierteren Term, welcher Ableitungen bezüglich der Orts- oder Impulskoordinate, x oder k , beinhaltet. Aus dieser Beobachtung können wir eine effiziente Strategie konzipieren, mit der wir das Problem numerisch lösen können:

Zunächst betrachten wir *nur* den multiplikativen Teil der Schrödingergleichung im Ortsraum

$$V(x)\psi(x,t) = i\hbar \frac{\partial}{\partial t} \psi(x,t).$$

Für einen Zeitschritt Δt nimmt die Integration also die Form

$$\psi(x,t + \Delta t) = \psi(x,t) e^{-iV(x)\Delta t/\hbar}$$

an. Transformieren wir nun die restliche Gleichung in den Impulsraum, so können hier wir den verbleibenden Teil

$$\frac{\hbar^2 k^2}{2m} \tilde{\psi}(k,t) = i\hbar \frac{\partial}{\partial t} \tilde{\psi}(k,t),$$

bearbeiten. Letzterer läuft auf eine Integration im Zeitschritt Δt der Form

$$\tilde{\psi}(k,t + \Delta t) = \tilde{\psi}(k,t) e^{-i\hbar k^2 \Delta t / 2m}$$

hinaus.

Numerische Lösung

Um das oben beschriebene Verfahren umzusetzen, müssen wir also wiederholte Male Fouriertransformierte berechnen. Dies ist ein zum einen kompliziertes, zum anderen aber auch ein extrem gut studiertes Problem. Der bei weitem effizienteste Algorithmus für die numerische Fourier-Transformation ist die sogenannte **Fast Fourier Transformation** (FFT), welcher in `scipy` zur Verfügung gestellt wird, und welchen wir hier verwenden wollen. Der Algorithmus in etwas konkreterer Form sieht also so aus (vgl. Abb. 2):

1. Wählen Sie a und b , und eine Schrittzahl N . Daraus ergibt sich $\Delta x = (b - a)/N$ und $\Delta k = 2\pi/(b - a)$. Wir definieren nun $x_n = a + n\Delta x$ und $k_m = k_0 + m\Delta k$, wobei $k_0 = -\pi/\Delta x$.
2. Diskretisieren Sie die Start-Wellenfunktion und das Potential auf diesem Gitter.
3. Um dieses System um einen Schritt Δt in der Zeit zu entwickeln, machen Sie folgendes:
 - a) Entwickeln Sie $\psi(x_n, t)$ um eine halbe Schrittweite:

$$\psi(x_n, t) \leftarrow \psi(x_n, t) \exp[-(i\Delta t/2)(V(x_n)/\hbar)] \quad (8)$$

- b) Berechnen Sie die Fouriertransformierte $\tilde{\psi}$.
- c) Entwickeln Sie $\tilde{\psi}(k_m, t)$ um einen ganze Schrittweite:

$$\tilde{\psi}(k_m, t) \leftarrow \tilde{\psi}(k_m, t) \exp[-i\hbar(k \cdot k)\Delta t/(2m)] \quad (9)$$

- d) Berechnen Sie ψ aus $\tilde{\psi}$ mittels der inversen Fouriertransformation.
- e) Berechnen Sie wieder einen halben Zeitschritt in ψ :

$$\psi(x_n, t) \leftarrow \psi(x_n, t) \exp[-(i\Delta t/2)(V(x_n)/\hbar)] \quad (10)$$

4. Wiederholen Sie den Prozess a) bis e) für jeden Zeitschritt.

Das Aufteilen der x -Zeitentwicklung in zwei halbe Schritte erweist sich als numerisch stabileres Verfahren, ähnlich wie beim Leap-Frog-Verfahren.

Kontinuierlich versus diskret: Beim Übergang von der kontinuierlichen zur diskreten Fourier-Transformation haben wir ein paar Vorfaktoren noch unbeachtet gelassen. Damit ihr Programm korrekt rechnet, muss die Funktion $\psi(x_n, t)$ vor der Fouriertransformation skaliert werden:

$$\psi(x_n, t) \rightarrow \frac{\Delta x}{\sqrt{2\pi}} \psi(x_n, t) e^{-ik_0 x}.$$

Mit dieser skalierten Funktion kann der Algorithmus nun durchgeführt werden. Wenn Sie die Funktion plotten wollen, muss diese Skalierung wieder rückgängig gemacht werden. Es bietet sich an, ein Extra-Array zu pflegen, welches nur zum Plotten verwendet wird, während die eigentlichen Berechnungen mit der skalierten Funktion geschehen.

Simulation

Nun noch einige Hinweise zu dem System, welches simuliert werden soll: Wieder geht es um ein Gaußsches Wellenpaket. Während wir allerdings in Aufgabe 25 ein freies Teilchen betrachtet haben, wollen wir nun eine Potentialbarriere errichten und die Streuung des Wellenpakets berechnen. Die Startwellenfunktion zum Zeitpunkt $t = 0$ ist also gegeben durch:

$$\psi(x, t = 0) = (\sigma\sqrt{\pi})^{-1/2} \exp\left[-\frac{1}{2}\left(\frac{x-x_0}{\sigma}\right)^2 + ixk_0\right]. \quad (11)$$

Wählen Sie folgende Werte:

$$\Delta x = 0.1, \quad N = 2^{11} = 2048, \quad \text{damit ergibt sich also } a = -102.4, b = 102.4$$

$$\Delta t = 0.01$$

$$x_0 = -30.0$$

$$k_0 = \sqrt{2m}$$

$$\sigma = 8/k_0.$$

Für das Potential verwenden Sie:

$$V(x) = \begin{cases} V_0 & 0 < x < 1 \\ 0 & \text{sonst.} \end{cases}$$

Plotten Sie $|\psi(x, t)|^2$ als Funktion der Zeit. Sie sollten dabei nicht jeden Schritt plotten (eher jeden 50.), da die Animation sonst zu langsam läuft.

Die Energie des Wellenpakets ist $E = 1$. Experimentieren Sie mit der Höhe V_0 des Potentials: Sie werden z.B. den **Tunneleffekt** finden, bei dem ein Teil der Wellenfunktion durch die Potentialbarriere *tunneln* kann, obwohl letztere eine höhere Energie hat. Die Wahrscheinlichkeit, dass das Teilchen hinter der energetisch höher liegenden Barriere auftaucht ist also nicht mehr Null — ein klassisch unmögliches Resultat. Finden Sie nun Werte für V_0 , bei denen **Resonanzen** auftreten, also ein Teil der Wellenfunktion innerhalb der Potentialbarriere “gefangen” bleibt. Untersuchen Sie dies sowohl für **repulsive** ($V_0 > 0$) als auch für **attraktive** ($V_0 < 0$) **Potentiale**.

Hinweis: Hier noch ein Beispiel, wie man die FFT verwendet:

```
from scipy.fftpack import fft, ifft
fk = fft(fx)
fx2 = ifft(fk)
```

Diese Beispiel transformiert die im NumPy-Array fx gespeichert Ortsraumfunktion in den Impulsraum und speichert das Ergebnis in fk . Danach wird diese Funktion wieder zurücktransformiert. fx und $fx2$ sollten nun die gleichen Werte enthalten.