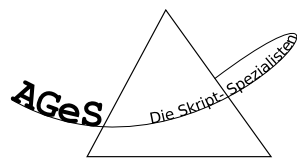


COMPUTATIONAL PHYSICS

nach den Vorlesungen von Dr. Arnd Bäcker
(Wintersemester 2008/09)

Herausgegeben von



Jeffrey Kelling
Felix Lemke
Stefan Majewsky

Stand: 22. Juni 2009

Inhaltsverzeichnis

Vorwort (zuerst lesen)	4
1 Einleitung	5
2 Abbildungen	6
2.1 Standardabbildung	6
3 Elementare numerische Methoden	7
3.1 Darstellung von Zahlen	7
3.2 Fehler in der Numerik	8
3.3 Ableitung	9
3.4 Nullstellensuche	9
3.4.1 Bisektionsmethode	10
3.4.2 Newton-Verfahren	10
3.5 Integration	10
3.6 Minimierung und Maximierung von Funktionen	12
4 Gewöhnliche Differentialgleichungen	14
4.1 Aufgabenstellung	14
4.2 Fehler	15
4.3 Schrittweitenanpassung	16
4.4 Einschritt-Verfahren	16
4.4.1 Euler-Verfahren	16
4.4.2 Runge-Kutta-Verfahren	17
4.4.3 Bulirsch-Stoer-Verfahren	17
4.5 Mehrschritt-Verfahren	17
4.6 Steife Differentialgleichungen	17
4.7 Physikalisches Beispiel: Elektronen in Halbleiternanostrukturen	18
4.7.1 Experiment	18
4.7.2 Einheitenlose Bewegungsgleichungen	18
5 Quantenmechanik in eindimensionalen Potentialen	20
5.1 Einheitenlose Schrödingergleichung	20
5.1.1 Harmonischer Oszillator	22
5.2 Lösungsverfahren für die zeitunabhängige Schrödingergleichung	22
5.2.1 Diskretisierungsmethode	22
5.2.2 Entwicklung nach Basisfunktionen	23
5.2.3 Rechenzeit und Speicherplatz	24
5.3 Zeitentwicklung von Wellenpaketen	25
5.4 Phasenraumdarstellung	26
5.4.1 Kohärenter Zustand	26
5.4.2 Husimi-Darstellung	27

5.4.3	Semiklassik	27
5.5	Räumlich periodische Systeme	28
5.5.1	Bloch-Theorem	28
5.5.2	Bandspektrum	28
6	Fourier-Transformation	29
6.1	Fourier-Reihen	29
6.2	Abtasttheorem	30
6.3	Diskrete Fourier-Transformation	30
6.4	Schnelle Fourier-Transformation (FFT)	31
7	Solitonen in der Physik	32
7.1	Korteweg-de-Vries-Gleichung	32
7.2	Numerik	33
	Stichwortverzeichnis	36

Vorwort

Bevor Ihr beginnt, mit diesem Skript zu arbeiten, möchten wir Euch darauf hinweisen, dass dieses Skript weder den Besuch der Vorlesung noch das selbstständige Nacharbeiten des Stoffes ersetzt. Wer das nicht verstanden hat, bei dem kann die Benutzung des Skriptes für Probleme insbesondere im Verständnis des Stoffes sorgen.

Das liegt daran, dass das Skript nicht als vorgekauter Wissensspeicher zu verstehen ist. Das hier ist eine Abschrift des Inhaltes, den die Vorlesung zu vermitteln versucht. Nicht enthalten sind zum Beispiel mündliche Kommentare des Professoren, auch wenn diese im individuellen Falle oft erst den Groschen fallen lassen.

Gut geeignet ist das Skript einfach gesagt als Wissensstütze, also zum Beispiel zum schnellen Nachschlagen; außerdem zum Wiederholen früheren Stoffes, sofern ein ausreichendes Grundverständnis vorhanden ist. Nach diesen einleitenden Worten wünschen wir Euch viel Spaß bei der Arbeit mit diesem Skript und viel Erfolg beim Studium!

Die AGeS-Redaktion
www.ages-skripte.org

P.S. Wir suchen immer Helfer, die unsere Skripte um neue Inhalte erweitern, Fehler suchen, oder das Layout ansprechender gestalten wollen. Wenn Ihr Lust habt, meldet Euch über unsere Webseite.

1 Einleitung

Computational Physics ist kein separates Teilgebiet der Physik, sondern eine Hilfsdisziplin, die die klassische Physik bei der Simulation und Visualisierung von Daten sowie bei numerischen Rechnungen unterstützt.

In dieser Vorlesung wird gezeigt, wie bekannte physikalische Probleme in Algorithmen gefasst werden können. Das Ziel ist:

- die Semantik der Physik besser zu verstehen
- die Intuition zu fördern (visualisierte Simulationen sind trivialerweise weitaus anschaulicher als mathematische Formeln)
- auch analytisch nicht lösbare Fälle untersuchen zu können (Bsp. Doppelpendel)

Weiterhin ergeben sich einige Nebenziele:

- Verständnis numerischer Verfahren
- physikalische Probleme einheitenlos machen (also die Reduktion komplexer Systeme durch Wegkürzen uninteressanter Einflüsse)
- Kampf mit endlicher Rechengenauigkeit sowie der diskontinuierlichen Wertebehandlung (z. B. iterative Algorithmen)
- kritisches Überprüfen der Ergebnisse

Auf der mathematisch-physikalischen Seite werden die folgenden Inhalte vermittelt:

- Abbildungen (reguläre und chaotische Dynamik)
- elementare Methoden der Numerik (z. B. Differenzieren, Integrieren, Nullstellenbestimmung)
- Bewegungsgleichungen
- Quantenmechanik (z. B. Spektren, Eigenfunktionen, Wellenpaketdynamik)
- evtl. Solitonendynamik (nichtlineare partielle Differentialgleichungen)
- stochastische Prozesse (z. B. Brownsche Bewegung)
- Fourier-Transformation

Die folgende Technikausstattung wird verwendet:

- Betriebssystem: Linux
- Editor: xemacs, Kate
- Programmiersprache: Python (manchmal auch: Mathematica, Maple, Matlab, IDL)

Zur Benotung: Auf jedem Übungsblatt gibt es einige Hausaufgaben, die bewertet werden (Abgabe per E-Mail bis zum nächsten Montag, 8:00). Um den Schein zu erhalten, muss man mindestens zwei Drittel der dadurch erreichbaren Punkte erhalten, sowie ein selbst gewähltes Simulationsprojekt implementieren und präsentieren.

2 Abbildungen

Wir beginnen mit einem Thema, das sich durch seine extrem einfache Numerik und eine trotzdem interessante Dynamik auszeichnet.

2.1 Standardabbildung

Das physikalische Modell ist ein **gekickter Rotor**: Wir betrachten eine Masse m an einem Stab der Länge r . Für einen freien Rotor ergibt sich die folgende Winkelbewegung:

$$\theta(t) = \theta_0 + \omega \cdot t$$

Dieses System hat einen konstanten Drehimpuls I und ein konstantes Trägheitsmoment mr^2 . Daraus ergibt sich die Winkelgeschwindigkeit $\omega = I/mr^2$. Nun wird dieser Rotor von rechts mit dem Impuls P periodisch zu den Zeiten $t = n \cdot T$ gekickt. Beim Kick ändert sich der Drehimpuls:

$$I' = I + rP \cdot \sin \theta \quad (A)$$

Zwischen den Kicks findet eine Winkeländerung statt:

$$\theta' = \theta + \omega \cdot T = \theta + \frac{I \cdot T}{mr^2} \quad (B)$$

In diesem System haben wir vier Parameter (r , m , P und T). Um das System zu reduzieren, führen wir **einheitenlose Variablen** ein.

- Der Winkel θ ist bereits einheitenlos.
- Ein typischer Drehimpuls ist $\frac{mr}{T} \cdot r = \frac{mr^2}{T}$. Damit lässt sich ein einheitenloser Drehimpuls *definieren*:

$$p = \frac{I}{\frac{mr^2}{T}} = \frac{IT}{mr^2}$$

Die entsprechenden **einheitenlosen Gleichungen** erhalten wir, indem wir (A) durch den typischen Drehimpuls dividieren, und in (B) den einheitenlosen Drehimpuls einsetzen:

$$\begin{aligned} p' &= p + \underbrace{\frac{rP \cdot T}{mr^2}}_{\equiv K} \cdot \sin \theta \\ \theta' &= \theta + p \end{aligned}$$

Das System hat also nur einen echten Parameter, die einheitenlose Kickstärke $K = PT/rm$. Wir betrachten nun θ und p jeweils kurz nach dem n -ten Kick:

$$\begin{aligned} \theta_{n+1} &= (\theta_n + p_n) \bmod 2\pi \\ p_{n+1} &= p_n + K \cdot \sin \theta_{n+1} \end{aligned}$$

3 Elementare numerische Methoden

3.1 Darstellung von Zahlen

Jede reelle Zahl kann man als Binärzahl darstellen:

$$x = \pm (\alpha_n \cdot 2^n + \alpha_{n-1} \cdot 2^{n-1} + \dots + a_0 \cdot 2^0 + \alpha_{-1} \cdot 2^{-1} + \dots)$$

Hierbei ist $\alpha_i \in \{0, 1\}$. Für **ganze Zahlen** (*Integer*) nimmt man meist 32 Bit (4 Byte), von denen ein Bit für das Vorzeichen reserviert wird. Damit kann man alle Zahlen zwischen -2^{31} und $2^{31} - 1 \approx 2 \cdot 10^9$ darstellen. Arithmetische Operationen auf solchen Zahlen sind immer exakt (abgesehen von Überläufen durch zu große oder zu kleine Operationsergebnisse).

Ein alternatives Konzept sind **Festkommazahlen**, bei denen man n_1 Binärziffern vor dem Komma sowie n_2 Nachkommaziffern speichert. Auch hier ist die Arithmetik exakt. Solche Zahlen eignen sich für Buchhaltungsrechnungen, sind aber zu inflexibel für die Anforderungen unserer Numerik.

Wir verwenden in den allermeisten Fällen **Gleitkommazahlen**. Diese werden oft in 64 Bit (8 Byte) gespeichert, die entsprechenden Datentypen sind *double* in C und *float* bzw. *numpy.float64* in Python. Die Gleitkommazahl teilt sich in drei Abschnitte:

$$x = (-1)^{\text{Vorzeichen}} \cdot \text{Mantisse} \cdot 2^{\text{Exponent}}$$

Die Größenverteilung dieser Abschnitte ist im IEEE-754-Standard festgelegt:

- Das Vorzeichen braucht 1 Bit im Speicher.
- 52 Bits sind für die Mantisse vorgesehen. Das bedeutet, dass es nur etwa 15 signifikante Ziffern gibt ($2^{52} \approx 5 \cdot 10^{15}$). Weitere Stellen werden abgeschnitten, was mitunter zu Rundungsfehlern führt. Eine Gleitkommazahl heißt **normalisiert**, wenn die erste Ziffer der Mantisse nicht Null ist. (Bei der Normalisierung ändert sich ggf. der Exponent, da die Mantisse um Zweierpotenzen verschoben wird, um das Normalisierungskriterium zu erfüllen.)
- In den letzten 11 Bits liegt der Exponent. Wegen $2^{11} = 2048$ ist der Exponent auf den Bereich zwischen -1022 und 1024 beschränkt. Wegen $2^{1024} \approx 10^{308}$ liegen die möglichen Zahlen im Bereich zwischen 10^{-308} und 10^{308} zzgl. der entsprechenden negativen Zahlen und der Null.

Die Arithmetik der Gleitkommazahlen ist aufgrund der begrenzten Länge der Mantisse nicht exakt. Zum Beispiel verschwinden zu kleine Zahlen bei der Addition:

$$1 + x = 1 \quad \text{für} \quad 0 \leq x < 2,22 \cdot 10^{-16}$$

Solche Ungenauigkeiten führen zu erstaunlichen Ergebnissen, die man bei der Erstellung von Algorithmen berücksichtigen muss, zum Beispiel:

$$(1 + \varepsilon_1) - (1 + \varepsilon_2) \neq \varepsilon_1 - \varepsilon_2 \quad \text{für kleine } \varepsilon_1, \varepsilon_2$$

Eine weitere Ungenauigkeit ergibt sich aus der Umwandlung aus dezimalen Eingaben. Ein Beispiel sei der folgende Python-Code:

```
x = 0.6
print repr(x)
```

Dieses Programm gibt 0,599 999 999 999 999 98 aus. Der fehlende Anteil von $2 \cdot 10^{-17}$ muss bei der Umwandlung in eine Gleitkommazahl abgeschnitten werden. Ein praktisches Problem damit ergibt sich, wenn ein Intervall $[a, b]$ in N Teilintervalle aufzuteilen ist. Dazu wird für jedes Teilintervall i der linke Randpunkt $l_i = a + i/N \cdot (b - a)$ errechnet:

```
step = 1.0 * (b-a) / N
x = a
while x < b:
    print x
    x = x + step
# Kürzer und äquivalent ist:
print arange(a, b, step)
```

Dies ergibt manchmal $N + 1$ Intervallgrenzen, weil $a + N \cdot [(b - a)/N] < b$ ist. Es gibt drei mögliche Lösungen für dieses Problem:

- Die Länge kann durch Nutzung sicherer Ganzzahlarithmetik sichergestellt werden:

```
print a + (b-a) * arange(N) / N
```

- Man kann die Ungenauigkeiten der Gleitkommaarithmetik durch andere Grenzwahl umschiffen:

```
print arange(a, b - step/2, step)
```

- *numpy* bietet eine alternative Funktion, die entsprechende Prüfungen beinhaltet:

```
print linspace(a, b, N, endpoint=False)
```

3.2 Fehler in der Numerik

- Bei Gleitkommazahlen sind **Rundungsfehler** unvermeidbar.
- Durch die numerische Lösung von Differentialgleichungen bzw. numerische Integration ergeben sich **Diskretisierungsfehler**.

Die Diskretisierungsfehler übersteigen meist die Rundungsfehler. Diese können durch geeignete Wahl des Algorithmus minimiert werden, was das eigentliche Ziel der Numerik ist. Der Worst Case sind **instabile Algorithmen**, bei denen die Rundungsfehler durch die Iterationen immer größer werden. Ein wichtiges praktisches Beispiel sind die Algorithmen der Besselfunktion (siehe *Numerical Recipes*, Kapitel 5.5).

3.3 Ableitung

Gegeben sei eine Funktion $f(x)$, zu der wir nun die Ableitung $f'(x)$ suchen. Möglich ist die Definition durch die Vorwärtsdifferenz:

$$f'_V(x) = \frac{f(x+h) - f(x)}{h} \quad \text{mit möglichst kleinem } h$$

Wie groß ist der Fehler, den wir durch diese Annäherung machen?

$$\begin{aligned} f'_V(x) - f'(x) &= \frac{1}{h} \cdot \left[f(x) + h \cdot f'(x) + \frac{h^2}{2} \cdot f''(x) + \dots \right] - \frac{1}{h} \cdot f(x) - f'(x) \\ &= \frac{f''(x)}{2} \cdot h + \dots = O(h) \end{aligned}$$

Der Fehler hängt linear von der Schrittweite h ab. Diesen Fehler kann man verkleinern, indem man stattdessen die Zentralsdifferenzmethode nutzt:

$$f'_Z(x) = \frac{f\left(x + \frac{h}{2}\right) - f\left(x - \frac{h}{2}\right)}{h}$$

Der Fehler lautet:

$$\begin{aligned} f'_Z(x) - f'(x) &= \frac{1}{h} \cdot \left[f(x) + \frac{h}{2} \cdot f'(x) + \frac{h^2}{2 \cdot 4} \cdot f''(x) + \frac{h^3}{6 \cdot 8} \cdot f'''(x) + \dots \right] \\ &\quad - \frac{1}{h} \cdot \left[f(x) - \frac{h}{2} \cdot f'(x) + \frac{h^2}{2 \cdot 4} \cdot f''(x) - \frac{h^3}{6 \cdot 8} \cdot f'''(x) + \dots \right] - f'(x) \\ &= \frac{f'''(x)}{24} \cdot h^2 + \dots = O(h^2) \end{aligned}$$

Welche Methode erlaubt es, einen relativen Fehler in der Größenordnung von 10^{-16} zu erreichen? Man würde erwarten, dass es mit beiden Methoden geht, wenn man $h \leq 10^{-8}$ für die Zentralsdifferenzmethode oder $h \leq 10^{-16}$ für die Vorwärtsdifferenzmethode nimmt.

Allerdings muss man in diesen Größenordnungen noch den numerischen Fehler beachten, der in der Größenordnung von $10^{-16}/h$ liegt. Der tatsächliche Fehler ergibt sich als Summe aus dem relativen Fehler des Verfahrens und dem numerischen Fehler. Das optimale h , also das h , für das der Gesamtfehler minimal wird, ist in etwa dann erreicht, wenn beide Fehler etwa gleich sind. Im Falle der Vorwärtsdifferenz ist:

$$\frac{10^{-16}}{h_{\min}} \stackrel{!}{=} h_{\min} \quad \Rightarrow \quad h_{\min} = 10^{-8}$$

Dies entspricht einem relativen Fehler von 10^{-8} . Für die Zentralsdifferenz ist der erreichbare relative Fehler 10^{-10} , denn:

$$\frac{10^{-16}}{h_{\min}} \stackrel{!}{=} h_{\min}^2 \quad \Rightarrow \quad h_{\min} \approx 10^{-5}$$

Die extrapolierte Differenz (siehe Übungsaufgaben) kann den relativen Fehler auf bis zu 10^{-12} senken.

3.4 Nullstellensuche

Wir suchen zu einer gegebenen Funktion $f(x)$ Nullstellen, also Lösungen x_i für $f(x_i) = 0$. Die mehrdimensionalen Fälle $f(\vec{x}_i) = 0$ und $\vec{f}(\vec{x}_i) = 0$ lassen wir außen vor, da diese Probleme sehr kompliziert werden. Im eindimensionalen Fall nutzen wir die Eigenschaft stetiger Funktionen, dass zwischen einer Stelle mit positivem Funktionswert und einer mit negativem Funktionswert immer mindestens eine Nullstelle liegt.

3.4.1 Bisektionsmethode

Diese Methode nutzt die obengenannte Eigenschaft stetiger Funktionen, um Nullstellen iterativ einzuschachteln:

- Die Startwerte a, b müssen Funktionswerte mit verschiedenen Vorzeichen haben ($f(a) \cdot f(b) < 0$).
- In jedem Iterationsschritt wird der Mittelwert $c = (a + b)/2$ gebildet. Ist $\text{sign } f(c) = \text{sign } f(a)$, so wird a durch c ersetzt. Ansonsten wird b durch c ersetzt.

Die Intervallbreite nach n Iterationen beträgt $(b - a) \cdot 2^{-n}$. Die bei Gleitkommazahlen kleinstmögliche Intervallbreite von $\approx 10^{-16}$ erreicht man also schon nach 50 Iterationen.

Die Funktion konvergiert linear gegen die Nullstelle. Wenn ε_n der Abstand zur Nullstelle im n -ten Schritt ist, so ist $\varepsilon_{n+1} = 0,5 \cdot \varepsilon_n$. Andere ähnliche Verfahren (Sekanten, Brent, ...) haben ein besseres Verhalten ($\varepsilon_{n+1} = \text{const.} \cdot \varepsilon_n^{1,618} \dots$).

Ein weiteres Problem dieses Verfahrens ist, dass doppelte Nullstellen (an denen der Graph die x -Achse nur berührt) oder nahe nebeneinander liegende Nullstellen oft übersprungen werden.

3.4.2 Newton-Verfahren

Dieses Verfahren nähert sich an die Nullstelle an, indem bei der aktuellen Position x_n eine Tangente an die Funktion gelegt wird. Deren Nullstelle x_{n+1} ist meist eine bessere Näherung:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Dieses Verfahren hat eine quadratische Konvergenz ($\varepsilon_{n+1} = \text{const.} \cdot \varepsilon_n^2$), also verdoppelt sich die Anzahl signifikanter Stellen in jedem Schritt. Allerdings benötigt das Newton-Verfahren die Kenntnis der Ableitung (eine numerische Ableitung vergrößert natürlich die Fehler). Außerdem kann bei schlechter Wahl des Startwertes das Newton-Verfahren divergent sein.

In der Praxis kombiniert man oft das Bisektionsverfahren mit dem Newton-Verfahren: Zunächst wird das Suchintervall mit Bisektion (oder dem Sekantenverfahren) eingeschränkt, dann wird dieses Intervall mit zwei bis drei Newton-Schritten verfeinert („root polishing“).

3.5 Integration

Das Ziel ist es den Ausdruck wie $I(a, b) = \int_a^b f(x) dx$ numerisch auszuwerten. Dazu ziehen wir die Definition des Riemannintegrals heran:

$$I(a, b) = \lim_{\substack{N \rightarrow \infty \\ \max_i (\Delta x_i) \rightarrow 0}} \sum_{i=1}^N f(\xi_i) \cdot \Delta x_i$$

Hierbei sind ξ_i beliebige Punkte im i -ten Intervall der Breite Δx_i . Beim Übergang in die Numerik wird N endlich, und man wählt meist den einfachen Fall:

$$h = \Delta x_i = \frac{b - a}{N} \quad \Rightarrow \quad x_i = a + i \cdot h \quad \text{mit} \quad i = 1, \dots, N$$

Wir bezeichnen mit $f_i = f(x_i)$ die Funktionswerte an diesen Stellen. (Stattdessen kann man auch Zwischenstellen z. B. mit $i = 1/2, 3/2, \dots$ nehmen.) Das Vorgehen wird von den **Newton-Cotes-Formeln** beschrieben:

- Betrachte Intervalle der Länge h .
- Approximiere die Funktion in diesen Intervallen durch Polynome.
- Leite die Formel für das Intervall h her.
- Summiere die Ergebnisse für alle Teilintervalle auf, um $I(a, b)$ zu erhalten.

Zu betrachten sind zwei Typen von Formeln:

- abgeschlossene Formeln, die $f(a)$ und $f(b)$ verwenden
- offene Formeln, die die Randpunkte nicht einbeziehen

Der Unterschied ist zum Beispiel relevant, wenn an den Rändern Singularitäten vorliegen.

1. Bei der **Trapezregel** (lineare Approximation) gilt für ein Teilintervall:

$$I(x_0, x_1) = h \cdot \left(\frac{f_0}{2} + \frac{f_1}{2} \right) + O(h^3 \cdot f')$$

Beim Übergang zum Integral zwischen a und b erhalten wir:

$$I(a, b) = h \cdot \left(\frac{f_0}{2} + f_1 + \dots + f_{N-1} + \frac{f_N}{2} \right) + O(\underbrace{N \cdot h^3}_{h^2 \cdot (b-a)} \cdot f'')$$

Wir wollen nun der Fehler zu $\int_a^b f(x) \, dx$ bestimmen. Da die Approximation für eine lineare Funktion exakt ist, wird der nächste Beitrag durch die quadratische Approximation zu f geliefert, das heißt: Der Fehler zum richtigen Integral entspricht in erster Näherung der Differenz zwischen der quadratischen Funktion und der linearen Näherung:

$$\Delta = I(x_0, x_1) - \int_{x_0}^{x_1} f(x) \, dx \approx \int_0^h -\frac{f''}{2} \cdot x \cdot (x - h) \, dx = \frac{f''}{2} \cdot \int_0^h (-x^2 + x \cdot h) \, dx = \frac{f'' \cdot h^3}{12}$$

Wie addiert sich der Fehler?

- Bei gleichem Vorzeichen der Krümmung addieren sich die Fehler.
- Bei sehr vielen Änderungen der Krümmung kann der Fehler wie \sqrt{N} anwachsen.

Man beachte weiterhin, dass bei der Summation $\sum_{i=0}^{N-1} I(x_i, x_{i+1})$ doppelt so viele Funktionsaufrufe nötig sind.

2. Ebenso gut und sogar einfacher ist die **Mittelpunktregel** für offene Formeln:

$$\begin{aligned} I(x_0, x_1) &= h \cdot f_{1/2} + O(h^3 \cdot f'') \\ I(a, b) &= h \cdot (f_{1/2} + f_{3/2} + \dots + f_{N-1/2}) + O(h^2) \end{aligned}$$

3. Die Berücksichtigung des quadratischen Verhaltens führt auf die **Simpson-Regel**:

$$\begin{aligned} I(x_0, x_1) &= h \cdot \left(\frac{f_0}{6} + \frac{4}{6} \cdot f_{1/2} + \frac{f_1}{6} \right) + O(h^5 \cdot f^{(4)}) \\ I(a, b) &= h \cdot \left(\frac{f_0}{6} + \frac{4}{6} \cdot f_{1/2} + \frac{2}{6} \cdot f_1 + \frac{4}{6} \cdot f_{3/2} + \frac{2}{6} \cdot f_2 + \dots + \frac{4}{6} \cdot f_{N-1/2} + \frac{f_N}{6} \right) \end{aligned}$$

Zur Herleitung: Das Ziel war die Approximation durch ein Polynom in der Form:

$$\int_0^1 p_i(x) \, dx = a \cdot p_i(0) + b \cdot p_i(1/2) + c \cdot p_i(1) \quad \text{mit} \quad p_i \in \{1, x, x^2\}$$

Dies liefert drei Gleichungen für die Unbekannten a , b und c .

Den Fehler für dieses Verfahren schätzt man durch die Differenz zum Polynom vierten Grades ab. Der Betrag von $f^{(3)}$ muss durch f_0 , $f_{1/2}$ und f_1 gehen, also muss der kubische Term an diesen Stellen verschwinden. Er ist somit um $h/2$ symmetrisch und fällt im Integral letztendlich weg. Auch der Betrag von $f^{(4)}$ muss durch f_0 , $f_{1/2}$ und f_1 gehen. Da für den x^4 -Term eine doppelte Nullstelle in $h/2$ möglich ist, fällt dieses Integral nicht weg:

$$\int_0^h O(x^4) dx = O(h^5 \cdot f^{(4)})$$

Bemerkenswert ist die Tatsache, dass eine quadratische Approximation bereits ein Skalierungsverhalten von $O(h^4)$ liefert. Auch interessant ist die folgende alternative Darstellung des Simpson-Integrals:

$$I_{\text{Simpson}}(h) = \frac{4}{3} \cdot I_{\text{Trapez}}\left(\frac{h}{2}\right) - \frac{1}{3} \cdot I_{\text{Trapez}}(h)$$

Hinter dem Wechsel zwischen $\frac{4}{6}$ und $\frac{2}{6}$ in der Simpsonregel steckt kein Geheimnis. In der gleichen Ordnung liegt eine andere Approximation:

$$I = \frac{h}{2} \cdot \left(\frac{9}{24} \cdot f_0 + \frac{28}{24} \cdot f_{1/2} + \frac{23}{24} \cdot f_1 + f_{3/2} + f_2 + \dots + f_{N-3/2} + \frac{23}{24} \cdot f_{N-1} + \frac{28}{24} \cdot f_{N-1/2} + \frac{9}{24} \cdot f_N \right) + O(h^4)$$

Der Aufwand der Methode liegt hauptsächlich in den Randintervallen.

4. Weitere Methoden ergeben sich durch Approximationen höherer Ordnungen, zum Beispiel:

- $n = 3$: $\frac{3}{8}$ -Regel
- $n = 4$: Milne-Regel mit $O(h^7)$
- $n = 6$: Weddie's Regel mit $O(h^9)$

5. Die **Romberg-Methode** berechnet $I(h)$, $I(h/2)$, $I(h/4)$ u. s. w. und extrapoliert zu $I(0)$.

6. Bei der **Gauß-Methode** werden die Stützstellen x_i an das Problem angepasst.

7. Man kann sich auch über die Mittel der Differentialrechnung dem Problem nähern:

$$I(a, b) = \int_a^b f(x) dx \quad \Rightarrow \quad \frac{dI}{db} = f(b) \quad \text{mit} \quad I(a, a) = 0$$

Dies ist eine Differentialgleichung mit Anfangswertproblem für I . Dieses Verfahren ist gut, falls man eine adaptive Schrittweite benötigt.

3.6 Minimierung und Maximierung von Funktionen

Das Maximierungsproblem von f entspricht dem Minimierungsproblem von $-f$. Wir konzentrieren uns deswegen im Folgenden auf die Minimierung. Zunächst schachteln wir das Minimum ein.

- Seien (a, b, c) so, dass $a < b < c$ sowie $f(b) < f(a)$ und $f(b) < f(c)$ gelten. Dann muss zwischen a und c ein Minimum vorliegen.
- Wähle zum Beispiel $x = (b+c)/2$. Ist $f(b) < f(x)$, fahre fort mit (a, b, x) , ansonsten mit (b, x, c) .

- Dies wird wiederholt, bis die Breite der Einschachtelung hinreichend klein ist.

Dieses Verfahren ist sicher, man findet also wirklich ein Minimum. Dies ist zum Beispiel bei den Nullstellensuchalgorithmen nicht der Fall, deswegen bietet sich für dieses Problem eine Kombination aus numerischer Ableitung und Nullstellensuche nur selten an.

Wie klein kann man die Einschachtelung machen? Man möchte idealerweise die numerische Genauigkeit erreichen:

$$(1 - \varepsilon) \cdot b < b < (1 + \varepsilon) \cdot b$$

Hierbei wäre $\varepsilon \approx 2,22 \cdot 10^{-16}$ die numerische Genauigkeitsgrenze. Dieser gute Fall ist allgemein nicht erreichbar, denn um das Minimum gilt:

$$f(x) \approx f(b) + \frac{1}{2} \cdot f''(b) \cdot (x - b)^2 = f(b) \cdot \left[1 + \frac{1}{2 \cdot f(b)} \cdot f''(b) \cdot (x - b)^2 \right]$$

Wenn der zweite Term in die Region von ε kommt, verschwindet er numerisch. Das passiert bei

$$|x - b| < \sqrt{\varepsilon} \cdot \sqrt{\frac{2 \cdot |f(b)|}{f''(b)}}$$

4 Gewöhnliche Differentialgleichungen

Die Physik steckt voller Differentialgleichungen, doch nur wenige davon sind analytisch lösbar. Bei der numerischen Lösung von Differentialgleichungen muss man immer die Frage betrachten, ob das sich eventuell ergebende Chaos eventuell numerisch bedingt ist. Unsere Ziele sind nun:

- Verständnis der Aufgabenstellung sowie der auftretenden Fehlerarten (lokal ggü. global, relativ ggü. absolut)
- grobe Kenntnis der genutzten Verfahren, um sie in Software nutzen zu können

4.1 Aufgabenstellung

Eine **Differentialgleichung** ist eine Gleichung mit unabhängigen Variablen (z. B. Zeit t) und abhängigen Variablen (z. B. Ort $x(t)$) sowie deren Ableitungen. Eine **gewöhnliche Differentialgleichung** enthält *nur eine* unabhängige Variable. Gibt es mehrere unabhängige Variablen, hat man eine **partielle Differentialgleichung**. Ein wichtiges Beispiel für eine partielle Differentialgleichung ist die Schrödingergleichung für $\psi(x, y, z, t)$.

Einige wichtige Beispiele für gewöhnliche Differentialgleichungen sind:

- Newtonsche Bewegungsgleichung, zum Beispiel: (mit Gewichtskraft und Reibung)

$$m \cdot \frac{d^2 z}{dt^2} = -mg + \eta \cdot \frac{dz}{dt}$$

- Pendel-Bewegungsgleichung:

$$ml \cdot \frac{d^2 \varphi}{dt^2} = -mg \cdot \sin \varphi$$

- Hamitonsche Bewegungsgleichungen:

$$\dot{\vec{q}} = \frac{\partial H(\vec{q}, \vec{p}, t)}{\partial \vec{p}} \quad \text{und} \quad \dot{\vec{p}} = -\frac{\partial H(\vec{q}, \vec{p}, t)}{\partial \vec{q}}$$

- allgemeine Differentialgleichung N -ter Ordnung:

$$\frac{d^N x}{dt^N} = F\left(t, x, \frac{dx}{dt}, \frac{d^2 x}{dt^2}, \dots, \frac{d^{N-1} x}{dt^{N-1}}\right)$$

Eine solche Gleichung kann man auf N Gleichungen erster Ordnung reduzieren, indem man die Hilfsvariablen $x_i(t) = \frac{d^i x}{dt^i}$ für $i = 1, \dots, N-1$ einführt. Dann gilt:

$$\begin{aligned} \dot{x} &= x_1 \\ \dot{x}_1 &= x_2 \\ &\vdots \\ \dot{x}_{N-2} &= x_{N-1} \\ \dot{x}_{N-1} &= F(t, x, x_1, \dots, x_{N-1}) \end{aligned}$$

Die für die Numerik relevante Form ist also eine reduzierte Differentialgleichung erster Ordnung mit M abhängigen Variablen:

$$\frac{dx_i(t)}{dt} = f_i(t, x_1, \dots, x_M) \quad \text{mit} \quad i = 1, \dots, M$$

Das zu lösende Problem wird vervollständigt durch

- das Anfangswertproblem: Zur Zeit t_0 sind die Anfangswerte $x_1(t_0), \dots, x_M(t_0)$ gegeben.
- das Randwertproblem: Die Startwerte von x_i sind zu verschiedenen Zeiten gegeben.

Im folgenden betrachten wir ein Anfangswertproblem mit $M = 1$. (Die Erweiterung auf $M > 1$ ist dann einfach.) Wir haben also die Form:

$$\frac{dy}{dt} = f(t, y) \quad \text{mit} \quad y(t_0) = y_0$$

Das Ziel ist die Lösung der Differentialgleichung zu diskreten Zeiten t_1, t_2, \dots . Es geht uns aber nicht um die exakten Werte $y(t_1), y(t_2), \dots$, diese sind realistisch nicht zu erreichen. Stattdessen suchen wir genäherte Werte y_1, y_2, \dots .

4.2 Fehler

- Rundungsfehler lassen sich durch Erhöhung der Genauigkeit reduzieren, was sich jedoch stark negativ auf die Rechengeschwindigkeit auswirkt, wenn man die native Präzision überschreiten will. Ein Test auf Rundungsfehler ist mittels Durchführung der Rechnung mit *geringer* Genauigkeit möglich.
- Analog zur Integration treten auch hier verfahrensabhängige Diskretisierungsfehler auf:

GRAPHIK

(wird nachgereicht)

- Sei $u(t_i) = y_i$ und $\frac{du}{dt} = f(t, u)$ ein Anfangswertproblem, dass in (t_i, y_i) beginnt. Löst man diese Differentialgleichung, so gelangt man zu $u(t_{i+1}) \neq y_{i+1}$. Dieser Unterschied wird als **lokaler Fehler** $\Delta_{\text{lokal}} = u(t_{i+1}) - y_{i+1}$ bezeichnet.

Dieser Fehler lässt sich nur schwer abschätzen, da $u(t_{i+1})$ nicht bekannt ist. Mann kann aber den Schritt $h = t_{i+1} - t_i$ mit zwei kleineren Schritten vergleichen:

$$\Delta_{\text{Lokal}} \approx \left| y_{i+1}(h) - y_{i+1} \left(\frac{h}{2}, \frac{h}{2} \right) \right|$$

- Der **globale Fehler** $\Delta_{\text{global}} = y(t_{i+1}) - y_{i+1}$ stellt in gewisser Weise die Summation über alle lokalen Fehler dar, enthält also alle Fehler, die seit dem Startwert (t_0, y_0) durch das numerische Verfahren approximiert worden sind. Häufig nimmt man für den globalen Fehler an:

* Im schlimmsten Fall sind die lokalen Fehler additiv: $\Delta_{\text{global}} = N \cdot \Delta_{\text{lokal}}$

* Im besten Fall sind die Fehler zufällig verteilt: $\Delta_{\text{global}} = \sqrt{N} \cdot \Delta_{\text{lokal}}$

Diese Annahmen sind falsch: Die lokalen Fehler verstärken sich gegenseitig, sodass der globale Fehler insb. in chaotischen Systemen exponentiell wächst:

$$\Delta_{\text{global}} = e^{\lambda \cdot t} \cdot \Delta_{\text{lokal}}$$

Hierbei ist λ der Liapunov-Faktor.

Wir sehen, dass der globale Fehler systemabhängig und nicht vorhersagbar ist. Um dennoch die Güte einer Rechnung zu prüfen, kann wie folgt vorgehen:

1. Wiederholung der Rechnung mit kleinerer Schrittweite, dabei das h solange verkleinert bis die Ergebnisse Stabil bleiben.
2. Die Rechnung rückwärts durchführen und prüfen, ob man wieder bei y_0 ankommt.

4.3 Schrittweitenanpassung

Teilweise ist eine kleine Schrittweite nötig, jedoch wäre es meist Verschwendung, sie überall beizubehalten, daher sollte man die Schrittweite dynamisch anpassen. Als Kriterium vergleicht man den lokalen Fehler mit einer externen Vorgabe Δ_{extern} für den maximalen lokalen Fehler pro Schritt. Dies erfordert zusätzliche Rechenzeit (z. B. 11/3 des Aufwands für 4. Ordnung des Runge-Kutta-Verfahrens). Ist tatsächlich $\Delta_{\text{lokal}} > \Delta_{\text{extern}}$, wird die Schrittweite h verkleinert und der Schritt wiederholt. Ist hingegen $\Delta_{\text{lokal}} < \Delta_{\text{extern}}$, so wird nach der Durchführung des aktuellen Schrittes die Schrittweite h vergrößert.

Die neue Schrittweite ergibt sich nach der folgenden Vorschrift:

$$h_{\text{neu}} = h \cdot \left(\frac{\Delta_{\text{extern}}}{\Delta_{\text{lokal}}} \right)^{1/n}$$

Hierbei entspricht n der Ordnung $O(h^n)$ des gewählten Verfahrens. Zum Beispiel ist $n = 5$ für das Runge-Kutta-Verfahren 4. Ordnung. Die Vorgabe des lokalen Fehlers ergibt sich wie folgt:

$$\Delta_{\text{extern}} = \varepsilon_{\text{absolut}} + \varepsilon_{\text{relativ}} \cdot |y_i|$$

Dabei sind $\varepsilon_{\text{relativ}}$ und $\varepsilon_{\text{absolut}}$ als Parameter übergeben. Den absoluten Anteil braucht man, weil y auch nahe gegen Null gehen kann. Der relative Fehler ist nötig, weil y auch sehr groß werden kann.

Vorsicht: Schlechte Routinen setzen $\varepsilon_{\text{absolut}} = \varepsilon_{\text{relativ}}$. Zu beachten ist weiterhin, dass die Einhaltung des vorgegebenen Fehlers nichts über den globalen Fehler aussagt.

4.4 Einschnitt-Verfahren

Unser Ziel ist das Auffinden einer Approximation zu $y(t)$ für $dy/dt = f(t, y)$ mit $y(t_0) = y_0$. Dabei schließen wir jeweils aus t_i, y_i auf t_{i+1}, y_{i+1} .

4.4.1 Euler-Verfahren

Der Schritt wird entlang der Tangente an der Kurve $y(t)$ im Punkt (t_i, y_i) durchgeführt:

$$f(t, y) = \frac{dy(t)}{dt} = \frac{y(t+h) - y(t)}{h} + O(h) \quad \Rightarrow \quad y_{i+1} = y_i + h \cdot f(t_i, y_i) + O(h^2)$$

Dies entspricht einer Taylor-Entwicklung erster Ordnung. Leider erzeugt dieses Verfahren viel zu große Fehler, außerdem ist das Verfahren instabil, wie zum Beispiel im Bild rechts dargestellt.

GRAPHIK

(wird nachgereicht)

4.4.2 Runge-Kutta-Verfahren

Wir beginnen mit dem Verfahren zweiter Ordnung („Mittelpunktmethode“). Dabei wird Information aus der Mitte des Intervalls für den Schritt über h verwendet, indem zunächst mithilfe des Eulerverfahrens der Schritt zu $t_i + h/2$ gemacht wird, und der dortige Anstieg genutzt wird.

$$\begin{aligned} k_1 &= h \cdot f(t_i, y_i) \\ k_2 &= h \cdot f\left(t_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right) \\ y_{i+1} &= y_i + k_2 + O(h^3) \end{aligned}$$

Eine noch bessere Approximation stellt die vierte Ordnung des Runge-Kutta-Verfahrens dar: Hierbei werden zunächst k_1 und k_2 wie beim RK-Verfahren zweiter Ordnung berechnet, dann wird der halbe Schritt in zweiter Ordnung wiederholt, und mit dem entstehenden k_3 ein ganzer Schritt gemacht:

$$\begin{aligned} k_3 &= h \cdot f\left(t_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right) \\ k_4 &= h \cdot f(t_i + h, y_i + k_3) \\ y_{i+1} &= y_i + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + O(h^5) \end{aligned}$$

4.4.3 Bulirsch-Stoer-Verfahren

Dieses Verfahren ist recht neu (etwa von 1980) und wird von *Numerical Recipes* favorisiert. Die Grundidee besteht darin, die Berechnung mit kleinen Schrittzahlen ($n = 2, 4, 6, \dots$) mit einer modifizierten Form der Mittelpunktmethode zweiter Ordnung durchzuführen. Die entstehende Folge $(y_i)_n$ von erreichten Endpunkten kann dann mithilfe von Polynomen oder rationalen Funktionen auf $n \rightarrow \infty$ extrapoliert werden.

4.5 Mehrschritt-Verfahren

Zur Lösung von Gleichungen der Form $\dot{y}(t) = f(t, y(t))$ verwendet man folgende Iteration:

$$y_{i+1} = y_i + \int_{t_i}^{t_{i+1}} f(t, y(t)) dt$$

Die Idee ist nun, die $f(t, y(t))$ für das Intervall $[t_i, t_{i+1}]$ durch Extrapolation aus zurückliegenden t_j, y_j mit Polynomen p -ter Ordnung anzunähern. Dieses Prinzip wird unter anderen in den Verfahren nach Adams und Barkforth, Adam und Moulton und beim Prädiktor-Korrektor-Verfahren eingesetzt. Auch hier ist wieder eine dynamische Schrittweitenanpassung nötig.

4.6 Steife Differentialgleichungen

Eine Differentialgleichung heißt **steif**, wenn deutlich unterschiedliche Zeitskalen vorkommen. Ein solches Verhalten beobachtet man häufig in der Physik:

- In einem Molekül bewegen sich die Atome relativ langsam, die Elektronen hingegen sehr schnell.

- Proteine in wässriger Lösung strukturieren sich langsam um, während die umgebenden Wassermoleküle sehr schnell ihre Wasserstoffbrückenbindungen umbauen.

Entsprechende Methoden sind die von Gear, oder die BDF („Backward Differentiation Formulas“).

4.7 Physikalisches Beispiel: Elektronen in Halbleiternanostrukturen

4.7.1 Experiment

Wir betrachten Grenzflächen von GaAs und AlGaAs. Letzteres ist mit Silizium-Donatoren dotiert. An der Grenzfläche befindet sich ein zweidimensionales Elektronengas mit einer reduzierten Elektronenmasse von $m^* \approx 0,067 \cdot m_e$ und einer Experimentell variierten Elektronendichte von $n \approx 1 \cdot 10^{15} \text{ m}^{-2}$. Daraus ergeben sich:

- Fermi-Wellenvektor $k_F = \sqrt{2\pi \cdot n}$
- Fermi-Energie $E_f = \frac{\hbar^2 \cdot k_F^2}{2m^*}$

Für die Leitfähigkeit sind die Elektronen nahe der Fermieenergie wichtig.

Die Dynamik der Elektronen wird durch folgende Parameter definiert:

- Die freie Weglänge liegt bei über $10 \mu\text{m}$, da es kaum Störstellen in der Atomgitterstruktur gibt.
- Die Strukturierung, z.B. Antidot-Gitter, spielt eine wesentliche Rolle. Ein **Dot** ist dabei ein Potentialtopf, meist verursacht durch einen Atomkern, an dem die Elektronendichte erhöht ist. **Antidots** sind hingegen Potentialberge, an denen die Elektronendichte verringert ist.

Legt man ein Magnetfeld an, so findet man in der Elektronendichteverteilung $\varrho(B)$ viele steile Antidots, sowie einige weiche (und stärkere) Antidots. Naiv erklärt man sich dies durch Zyklotronorbits, die nicht zum Strom beitragen und somit den Widerstand gegenüber dem kommensurablen Magnetfeld erhöht. Allerdings gibt es einen Peak, der einem Orbit um sechs bis sieben Antidots entspräche, was geometrisch nicht möglich ist. Dieser Peak lässt sich also nicht mithilfe des Antidot-Gitters erklären, man muss stattdessen die Dynamik untersuchen.

Problematischerweise liegt bei der Messung auch eine Spannung an, daher gibt es eigentlich keine Zyklotronorbits, sondern einen $\vec{E} \times \vec{B}$ -Drift. Nur wenn das angelegte elektrische Feld nicht zu groß ist, bleiben die Zyklotronorbits für kurze Zeiten erhalten.

Die Lösung des Problems ist durch das KAM-Theorem gegeben: Die Zyklotronorbits bleiben erhalten und tragen zum Peak bei. Der Hauptbeitrag zum Peak ergibt sich aber aus Trajektorien, die zeitweise in der Nähe der Zyklotronorbits hängen.

4.7.2 Einheitenlose Bewegungsgleichungen

Die Hamilton-Funktion für ein einzelnes Elektron lautet:

$$H(\vec{r}, \vec{p}) = \frac{(\vec{p} - e \cdot \vec{A})^2}{2m^*} + V(x, y) \quad \text{mit} \quad \vec{A} = \frac{B}{2} \cdot \begin{pmatrix} -y \\ x \\ 0 \end{pmatrix}$$

Diese Gleichung hat vier Parameter (e , m^* , B und E_F). Nun führen wir einheitenlose Größen ein:

- typische Länge: Antidot-Gitterperiode a , somit $\tilde{x} = x/a$ und $\tilde{y} = y/a$
- typische Energie: Fermi-Energie E_F , damit $\tilde{H} = H/E_F$ und $\tilde{V}(\tilde{x}, \tilde{y}) = \frac{V(x,y)}{E_F}$
- typischer Impuls: $\sqrt{m^* \cdot E_F}$ (aus $H = p^2/2m^*$), damit $\tilde{\vec{p}} = \vec{p}/\sqrt{m^* \cdot E_F}$
- typische Zeit:

$$\frac{\text{Länge}}{\text{Impuls pro Masse}} = \frac{a}{\sqrt{m^* \cdot E_F}/m^*} = \sqrt{\frac{a^2 \cdot m^*}{E_F}} = t_0 \quad \Rightarrow \quad \tilde{t} = \frac{t}{t_0}$$

- typisches Magnetfeld: B_0 so, dass der Zyklotrondurchmesser a ist

$$r_c^0 = \frac{a}{2} = \frac{\hbar \cdot k_F}{e \cdot B_0} \quad \Rightarrow \quad B_0 = \frac{2\hbar \cdot k_F}{a \cdot e} = \frac{2 \cdot \sqrt{2m^* \cdot E_F}}{a \cdot e} \quad \Rightarrow \quad \tilde{B} = \frac{B}{B_0}$$

Unter Verwendung dieser Größen bestimmen wir nun die einheitenlose Hamiltonfunktion:

$$\begin{aligned} \tilde{H} &= \frac{H}{E_F} = \frac{(\vec{p} - e\vec{A})^2}{2m^* \cdot E_F} + \frac{V(x,y)}{E_F} \\ &= \frac{1}{2} \cdot \left(\frac{\vec{p}}{\sqrt{m^* \cdot E_F}} - \frac{e \cdot B a}{2\sqrt{m^* \cdot E_F}} \cdot \begin{pmatrix} -y/a \\ x/a \\ 0 \end{pmatrix} \right)^2 + \tilde{V}(\tilde{x}, \tilde{y}) \\ &= \frac{1}{2} \cdot \left(\tilde{\vec{p}} - \sqrt{2} \tilde{B} \cdot \begin{pmatrix} -\tilde{y} \\ \tilde{x} \\ 0 \end{pmatrix} \right)^2 + \tilde{V}(\tilde{x}, \tilde{y}) \end{aligned}$$

Wir lassen die Tilden weg und verwenden von nun an nur noch einheitenlose Größen. Insgesamt lautet unsere Hamilton-Funktion also:

$$H(p_x, p_y, x, y) = \frac{(p_x + \sqrt{2} \cdot B \cdot y)^2}{2} + \frac{(p_y - \sqrt{2} \cdot B \cdot x)^2}{2} + V(x, y)$$

Die Hamiltonschen Bewegungsgleichungen lauten:

$$\begin{aligned} \dot{x} &= \frac{\partial H}{\partial p_x} = p_x + \sqrt{2} \cdot B \cdot y \\ \dot{y} &= \frac{\partial H}{\partial p_y} = p_y - \sqrt{2} \cdot B \cdot x \\ \dot{p}_x &= -\frac{\partial H}{\partial x} = \sqrt{2} \cdot B \cdot (p_y - \sqrt{2} \cdot B \cdot x) - \frac{\partial V}{\partial x} \\ \dot{p}_y &= -\frac{\partial H}{\partial y} = -\sqrt{2} \cdot B \cdot (p_x + \sqrt{2} \cdot B \cdot y) - \frac{\partial V}{\partial y} \end{aligned}$$

Führen wir die Abkürzung $v_x = p_x + \sqrt{2} \cdot B \cdot y$ (und v_y analog mit Minus) ein, so verkürzen sich diese Gleichungen weiter:

$$\begin{aligned} \dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{p}_x &= 2\sqrt{2} \cdot B \cdot v_y - \frac{\partial V}{\partial x} \\ \dot{p}_y &= -2\sqrt{2} \cdot B \cdot v_x - \frac{\partial V}{\partial y} \end{aligned}$$

Als Modell für das Potential verwenden wir:

$$V(x, y) = V_0 \cdot [\cos(\pi x) \cdot \cos(\pi y)]^4$$

Experimentell liegt der Durchmesser der Antidots bei $E_F = 1$ etwa $a/3$, daraus folgt:

$$V_0 = \left(\frac{1}{\cos \frac{\pi}{6}} \right)^4$$

5 Quantenmechanik in eindimensionalen Potentialen

Unser Ziel ist das bessere Verständnis der Eigenschaften einfacher quantenmechanischer Systeme:

- Eigenwerte und Eigenfunktionen
- Zeitentwicklung
- quantenmechanische Phasenraumdarstellung

Die zeitabhängige Schrödingergleichung für Teilchen im eindimensionalen Potential lautet:

$$\begin{aligned} i\hbar \cdot \frac{\partial}{\partial t} \varphi(x, t) &= H \varphi(x, t) \\ H &= -\frac{1}{2m} \cdot \frac{\partial^2}{\partial x^2} + V(x) \end{aligned}$$

Dies ist eine partielle Differentialgleichung für die unabhängigen Variablen x und t sowie die abhängige Variable $\varphi = \varphi(x, t)$. Allerdings ist ihre Struktur sehr einfach, da sie linear und homogen in $\varphi(x, t)$ ist. Daher gilt das **Superpositionsprinzip**: Zu jedem Paar von Lösungen $\varphi_{1/2}(x, t)$ ist auch $c_1 \cdot \varphi_1(x, t) + c_2 \cdot \varphi_2(x, t)$ eine Lösung.

Der Hamiltonoperator ist zeitunabhängig, somit ist die Separation durch einen Produktansatz möglich:

$$\varphi_n(x, t) = e^{-i \cdot \frac{E_n}{\hbar} \cdot t} \cdot \psi_n(x)$$

Für den ortsabhängigen Anteil gilt die zeitunabhängige Schrödingergleichung:

$$H \psi_n(x) = E_n \cdot \psi_n(x)$$

Dabei ist E_n der n -te Energieeigenwert zur n -ten Eigenfunktion $\psi_n(x)$. Die allgemeine Lösung wird beschrieben durch die Superposition aller möglichen Eigenfunktionen:

$$\varphi(x, t) = \sum_n c_n \cdot e^{-i \cdot \frac{E_n}{\hbar} \cdot t} \cdot \psi_n(x)$$

5.1 Einheitenlose Schrödingergleichung

Um einheitenlose Variablen zu erhalten beziehen wir diese wie folgt auf typische Größen:

- typische Länge: a durch Potential bestimmt, damit $\tilde{x} = x/a$

- typische Energie: V_0 durch Potential bestimmt, damit $\tilde{E}_n = E_n/V_0$ und

$$\tilde{H} = \frac{H}{V_0} = -\frac{1}{2} \cdot \underbrace{\frac{\hbar^2}{m \cdot V_0 \cdot a^2}}_{=\hbar_{\text{eff}}^2} \cdot \frac{\partial^2}{\partial \tilde{x}^2} + \tilde{V}(\tilde{x})$$

Im Hamilton-Operator taucht ein einheitenloser Parameter auf:

$$\hbar_{\text{eff}} = \frac{\hbar}{a \cdot \sqrt{m \cdot V_0}}$$

Der Nenner in diesem Term ist die typische Klassische Wirkung (Länge a und Impuls $\sqrt{m \cdot V_0}$).

- typischer Impuls: $\sqrt{m \cdot V_0}$ aus $H = p^2/2m$, damit $\tilde{p} = \hat{p}/\sqrt{m \cdot V_0}$
- Eine typische Zeit leiten wir aus den obigen Größen ab:

$$\text{typische Zeit} = \frac{\text{Länge}}{\text{Impuls/Masse}} = \frac{a}{\sqrt{m \cdot V_0}/m} = \sqrt{\frac{a^2 \cdot m}{V_0}} \equiv t_0 \quad \Rightarrow \quad \tilde{t} = \frac{t}{\sqrt{\frac{a^2 \cdot m}{V_0}}}$$

Die einheitenlose zeitabhängige Schrödingergleichung lautet nun:

$$i \cdot \underbrace{\frac{\hbar}{V_0} \cdot \sqrt{\frac{V_0}{a^2 \cdot m}}}_{=\hbar_{\text{eff}}} \cdot \frac{\partial}{\partial \tilde{t}} \tilde{\varphi}(\tilde{x}, \tilde{t}) = \tilde{H} \tilde{\varphi}(\tilde{x}, \tilde{t})$$

Eine Umrechnungsvorschrift für φ und $\tilde{\varphi}$ erhält man aus der Normierung:

$$1 = \int_{-\infty}^{\infty} |\varphi(x, t)|^2 dx \stackrel{!}{=} \int_{-\infty}^{\infty} |\tilde{\varphi}(\tilde{x}, \tilde{t})|^2 d\tilde{x} \quad \Rightarrow \quad \tilde{\varphi}(\tilde{x}, \tilde{t}) := \sqrt{a} \cdot \varphi(x(\tilde{x}), t(\tilde{t}))$$

Unser Endergebnis lautet (nun werden die Tilden der Einfachheit halber wieder weggelassen):

$$\begin{aligned} i\hbar_{\text{eff}} \cdot \frac{\partial}{\partial t} \varphi(x, t) &= H \varphi(x, t) \\ H &= -\frac{\hbar_{\text{eff}}^2}{2} \cdot \frac{\partial^2}{\partial x^2} + V(x) \end{aligned}$$

Für die zeitunabhängige Schrödingergleichung sowie deren Lösung gilt entsprechend:

$$\begin{aligned} H \cdot \psi_n(x) &= E_n \cdot \psi_n(x) \\ \varphi(x, t) &= \sum_n c_n \cdot e^{-i \cdot \frac{E_n}{\hbar_{\text{eff}}} \cdot t} \cdot \psi_n(x) \end{aligned}$$

Bemerkung

Indem \hbar durch \hbar_{eff} ersetzt und m weggelassen wird, konnte das System auf nur einen Parameter reduziert werden: Es wird alleine durch \hbar_{eff} sowie die einheitenlose Potentialform beschrieben.

5.1.1 Harmonischer Oszillator

Wir vergleichen den harmonischen Oszillator mit Einheiten mit dem einheitenlosen Problem:

Mit Einheiten	Einheitenlos
$V(x) = \frac{m}{2} \cdot \omega^2 \cdot x^2 = \frac{V_0}{2} \cdot \left(\frac{x}{a}\right)^2$	$V(x) = \frac{1}{2}x^2$
Zuordnung $V_0/a^2 = m\omega^2$ nicht eindeutig: Wähle wegen Energieskala $V_0 = \hbar\omega$, damit $a = \sqrt{\hbar/m\omega}$.	$\hbar_{\text{eff}} = \frac{\hbar}{\sqrt{\frac{\hbar}{m\omega}} \cdot \sqrt{\hbar m\omega}} = 1$
$E_n = \hbar\omega \cdot \left(n + \frac{1}{2}\right)$	$E_n = n + \frac{1}{2}$
$\psi_n(x) = \sqrt{\frac{4}{\pi\hbar}} \cdot \frac{1}{\sqrt{2^n \cdot n!}} \cdot e^{-\frac{m\omega}{2\hbar}x^2} \cdot H_n\left(\sqrt{\frac{m\omega}{\hbar}} \cdot x\right)$	$\psi_n(x) = \frac{1}{\sqrt{4\pi \cdot \sqrt{2^n \cdot n!}}} \cdot e^{-x^2/2} \cdot H_n(x)$

In der Ortsdarstellung der Eigenfunktionen tauchen die **Hermite-Polynome** auf:

$$\begin{aligned} H_0 &= 1 \\ H_1 &= 2 \cdot x \\ H_2 &= 4 \cdot x^2 - 2 \\ H_{n+1} &= \left(2 \cdot x - \frac{d}{dx}\right) H_n(x) \end{aligned}$$

5.2 Lösungsverfahren für die zeitunabhängige Schrödingergleichung

Der harmonische Oszillator ist ein gutes Modell zum Testen numerischer Verfahren, da es analytisch vollständig gelöst ist. Eine Überprüfung der Korrektheit numerischer Ergebnisse kann anhand der Eigenwerte, Eigenfunktionen und der Normierung erfolgen.

Wir wollen nun einige numerische Verfahren diskutieren.

5.2.1 Diskretisierungsmethode

Man geht – wie in der Numerik immer nötig – in Schritten Δx vor, wobei $\Delta x \rightarrow 0$ auf das exakte Ergebnis führen würde. Als Näherung für die zweite Ortsableitung können wir ansetzen:

$$\frac{d^2}{dx^2}\psi(x) \approx \frac{\psi(x + \Delta x) - \psi(x) - (\psi(x) - \psi(x - \Delta x))}{(\Delta x)^2}$$

- Diese Näherung ist gut, wenn $\psi(x)$ glatt auf der Skala Δx ist.
- Da Eigenfunktionen höherer Energien stärker variieren ist bei solchen ein kleineres Δx notwendig.
- Es gibt bessere Näherungen, z.B. auf Grundlage Fouriertransformation.

Einsetzen in die zeitunabhängige Schrödingergleichung ergibt zunächst:

$$-\frac{\hbar_{\text{eff}}^2}{2(\Delta x)^2} \cdot [\psi(x + \Delta x) - 2\psi(x) + \psi(x - \Delta x)] + V(x) \cdot \psi(x) = E \cdot \psi(x)$$

Wir definieren:

$$\begin{aligned} \psi_i &= \psi(x_0 + i \cdot \Delta x) \\ V_i &= V(x_0 + i \cdot \Delta x) \\ z &= \frac{\hbar_{\text{eff}}^2}{2(\Delta x)^2} \end{aligned}$$

Damit geht die Schrödingergleichung in ein lineares Gleichungssystem über:

$$-z \cdot (\psi_{i+1} + \psi_{i-1}) + (V_i + 2z) \cdot \psi_i = E \cdot \psi_i \quad \forall i \in \mathbb{Z}$$

In Matrixform lautet dieses Gleichungssystem:

$$\begin{pmatrix} \ddots & \vdots & \vdots & \vdots & \cdots \\ \cdots & V_{i-1} + 2z & -z & 0 & \cdots \\ \cdots & -z & V_i + 2z & -z & \cdots \\ \cdots & 0 & -z & V_{i+1} + 2z & \cdots \\ & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \cdot \begin{pmatrix} \vdots \\ \psi_{i-1} \\ \psi_i \\ \psi_{i+1} \\ \vdots \end{pmatrix} = E \cdot \begin{pmatrix} \vdots \\ \psi_{i-1} \\ \psi_i \\ \psi_{i+1} \\ \vdots \end{pmatrix}$$

Diese Matrix ist für die numerische Auswertung aufgrund ihrer unendlichen Größe leicht unhandlich. Es lohnt daher, die Matrix an Stellen x_{\min} und x_{\max} abzuhacken, an denen die Wellenfunktion sicher schon verschwunden ist. Dies entspricht der Näherung für das Potential $V(x < x_{\min}) = V(x > x_{\max}) = \infty$. Wir nehmen also $\psi(x = x_{\min}) = \psi(x = x_{\max}) = 0$ an. Diese Näherung ist gut für Eigenfunktionen $\psi(x)$, die an den Grenzen schon genügend stark abgefallen sind. Da Eigenfunktionen höherer Energien breiter sind, ist für diese ein größeres Intervall $[x_{\min}, x_{\max}]$ nötig.

Für die Berechnung wählen wir nun $x_0 = x_{\min}$ und $\Delta x = (x_{\max} - x_{\min})/(N-1)$. Als Randbedingungen führen wir $\psi_0 = \psi_{N-1} = 0$ ein, und erhalten eine endliche Matrix-Gleichung

$$\begin{pmatrix} V_1 + 2z & -z & \cdots & 0 \\ -z & V_2 + 2z & -z & \vdots \\ & -z & V_3 + 2z & -z \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & -z & V_{N-1} + 2z \end{pmatrix} \cdot \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \vdots \\ \psi_{N-1} \end{pmatrix} = E \cdot \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \vdots \\ \psi_{N-1} \end{pmatrix}$$

Die Diagonalisierung der Matrix ergibt dann E_n und $\psi_{n,i}$ mit $n, i = 1, 2, \dots, N-2$. Hierzu verwendet man eine Routine zur numerischen Bestimmung von Eigenwerten und -vektoren einer Matrix. SciPy stellt hierfür die Methoden `scipy.linalg.eig(h)` (Variante für hermitesche Matrizen wie diese) bereit.

- Parameter der Numerik sind:

Diskretisierungsschritt Δx bzw. Schrittzahl N

Intervall $[x_{\min}, x_{\max}]$

- Diese Methode ist auf beliebige Potentiale anwendbar (ohne weitere Vorarbeit).
- Sie ebenfalls leichter auf mehrere Dimensionen erweiterbar (obwohl die Effizienz leidet).

5.2.2 Entwicklung nach Basisfunktionen

Sei $\{|\nu\rangle\}$ eine Orthonormalbasis. Mögliche Basen in Ortsdarstellung $f_\nu(x) = \langle x | \nu \rangle$ sind:

- ν -te Eigenfunktion eines harmonischen Oszillators: $f_\nu(x) \sim H_\nu(x) \cdot \exp(-x^2/2)$ mit $\nu = 0, 1, \dots$
- ν -te Eigenfunktion eines Kastenpotentials: $f_\nu(x) \sim \sin(\pi\nu \cdot x)$ mit $\nu = 1, 2, \dots$

Aus der zeitabhängigen Schrödingergleichung folgt:

$$\begin{aligned}\hat{H}|\psi_n\rangle &= E_n|\psi_n\rangle \\ \sum_{\nu'} \langle \nu' | \hat{H} | \nu \rangle \cdot \langle \nu' | \psi_n \rangle &= E_n \cdot \langle \nu' | \psi_n \rangle \\ \sum_{\nu'} H_{\nu'\nu} \cdot \psi_{n,\nu'} &= E_n \cdot \psi_{n,\nu'}\end{aligned}$$

Dabei ist $H_{\nu'\nu}$ die Matrixdarstellung des Hamiltonoperators:

$$H_{\nu'\nu} = \int_{-\infty}^{+\infty} f_{\nu'}^*(x) \left(-\frac{\hbar_{\text{eff}}^2}{2} \cdot \frac{d^2}{dx^2} + V(x) \right) \cdot f_{\nu}(x) dx$$

Diese Matrix ist unendlich groß. Um mit ihr überhaupt rechnen zu können, muss die Matrix bei einem endlichen ν abgeschnitten werden. Somit sind nur solche Lösungen gut, die durch die verbliebenen Basisfunktionen gut dargestellt werden können. Eine interessante physikalische Fragestellung ist die geeignete Wahl der Basis, sodass sich schon mit einer kleinen Basis eine genügend große Genauigkeit erzielt werden kann.

5.2.3 Rechenzeit und Speicherplatz

Meist muss man zwischen Geschwindigkeit und Speicherplatzbedarf einen Kompromiss finden. Dieser Kompromiss ist stark vom Problem abhängig:

- Was möchte man erhalten?
 - Eigenwerte (alle oder nur in einem bestimmten Intervall)
 - Eigenvektoren (alle oder nur wenige ausgewählte)
- Was für eine Matrix steht zur Verfügung?
 - Welche Symmetrien liegen vor (symmetrische, hermitesche, orthogonale, unitäre Matrix)?
 - Welche Form hat die Matrix (tridiagonal, Herzenberg, Band, dünn/dicht besetzt)?

Entsprechend der Anforderungen kann man aus verschiedenen Software-Paketen (LAPACK, IMSC, NAG) wählen. Die relevante Komponente für unsere Anwendungen ist das BLAS (*Basic Linear Algebra Subprogram*) von LAPACK, welches von SciPy verwendet wird.

Wie verhält sich der Rechenaufwand zur Bestimmung von Eigenwerten unter Ausnutzung der Matrixgestalt?

- Kann man keine Symmetrie annehmen, liegt der Rechenaufwand ungefähr bei $O(N^3)$, eine typische Praktikabilitätsgrenze ist $N_{\text{max}} \approx 10^4$. Eine solche Matrix braucht im Speicher schon $10^4 \cdot 10^4 \cdot 16 \text{ Bytes} \approx 1,6 \text{ GB}$, was gerade noch vertretbar ist.
- Liegt eine tridiagonale Symmetrie vor, so reduziert sich der Rechenaufwand auf $O(N^2)$, sodass Matrizen bis $N_{\text{max}} \approx 10^6$ diagonalisiert werden können. Der Speicheraufwand für die Matrix ist auch dann noch vertretbar, allerdings können aus Platzgründen nicht alle Eigenvektoren ermittelt werden. Stattdessen bestimmt man einzelne Eigenvektoren.
- Im Falle einer Bandmatrix mit der Bandbreite b liegt der Rechenaufwand bei $O(bN^2)$.

5.3 Zeitentwicklung von Wellenpaketen

Seien \hat{H} zeitunabhängig ($\partial\hat{H}/\partial t = 0$) und dessen Eigenfunktionen $\{\psi_n(x)\}$ sowie dessen Eigenenergien $\{E_n\}$ bereits bekannt. Dann lässt sich ein beliebiger Zustand als Superposition der Eigenzustände darstellen:

$$\varphi(x, t) = \sum_n c_n \cdot e^{-iE_n \cdot t/\hbar} \cdot \psi_n(x)$$

Im einheitenlosen Fall wird \hbar zu \hbar_{eff} . Unser Ziel ist die Berechnung der Anregungskoeffizienten c_n . Für $t = 0$ gilt zunächst:

$$\varphi_0(x) \equiv \varphi(x, t = 0) = \sum_n c_n \cdot \psi_n(x)$$

Nun nutzen wir die Orthogonalität der Eigenfunktionen:

$$\int_{-\infty}^{\infty} \psi_m^*(x) \cdot \varphi_0(x) \, dx = \sum_n c_n \cdot \underbrace{\int_{-\infty}^{\infty} \psi_m^*(x) \cdot \psi_n(x) \, dx}_{=\delta_{m,n}} = c_m$$

Die Eigenfunktionen, wie auch das Anfangswellenpaket, müssen normiert sein:

$$\int_{-\infty}^{+\infty} |\psi_n(x)|^2 \, dx = 1 \quad \Rightarrow \quad \sum_{i=0}^{N-1} \left| \psi_0^{\text{numerisch}}(x_i) \right|^2 \cdot \Delta x \stackrel{!}{=} 1$$

Letzteres ist die bestmögliche Näherung, falls $\psi_n(x)$ in der Nähe des Randes sehr klein ist. Für die Anregungskoeffizienten folgt jetzt numerisch:

$$c_n^{\text{numerisch}} = \Delta x \cdot \sum_{i=0}^{N-1} \psi_n^*(x_i) \cdot \varphi_0(x_i)$$

Niederenergetische Eigenfunktion sind numerisch gut daher sind auch die entsprechenden c_n nutzbar. Die zu den Eigenfunktionen höherer Energie gehörigen und nicht verwendbaren $c_n^{\text{numerisch}}$ sollten sehr klein sein. Man sollte also immer testen, ob in der Lösung die c_n schnell abfallen. In Python sieht dies wie folgt aus:

```
ew, ev = numpy.linalg.eigh(matrix)
c = dx * dot(conjugate(transpose(ev)), phi0)
```

Zum Testen der Zeitenwicklung kann man wieder ein analytisch gelöstes Problem wie den harmonischen Oszillator nehmen. In diesem bleibt ein anfängliches Gauß-Wellenpaket immer ein Gauß-Wellenpaket, lediglich die Breite variiert.

Allgemeine Eigenschaften der Zeitentwicklung

Die Wellenpakete bleiben immer normiert:

$$1 = \int_{-\infty}^{\infty} |\varphi(x, t)|^2 \, dx = \int_{-\infty}^{\infty} \left(\sum_m c_m^* \cdot e^{iE_m \cdot t/\hbar} \cdot \psi_m^*(x) \right) \cdot \left(\sum_n c_n \cdot e^{-iE_n \cdot t/\hbar} \cdot \psi_n(x) \right) \, dx = \sum_n |c_n|^2$$

Der Energieerwartungswert ist zeitunabhängig:

$$\begin{aligned}
 \langle E(t) \rangle &= \langle \varphi(t) | H | \varphi(t) \rangle \\
 &= \int_{-\infty}^{+\infty} \varphi^*(x, t) \cdot H(x, t) \cdot \varphi(x, t) \, dx \\
 &= \int_{-\infty}^{+\infty} \left(\sum_m c_m^* \cdot e^{iE_m t/\hbar} \cdot \psi_m^*(x) \right) \cdot \left(\sum_n c_n \cdot e^{-iE_n t/\hbar} \cdot E_n \cdot \psi_n(x) \right) \, dx \\
 &= \sum_n |c_n|^2 \cdot E_n
 \end{aligned}$$

Auch den Energieeigenwert berechnet man mit Python in einer Zeile:

```
energy = dot(abs(c) ** 2, ew)
```

Eine andere interessante Frage ist die nach dem Zeitmittel des Wellenpaketes. Startet zum Beispiel ein Wellenpaket in der linken Mulde eines Doppelmuldenpotentials, wo ist es im langfristigen Mittel?

$$\left\langle |\varphi(x, t)|^2 \right\rangle_t = \left\langle \sum_{m,n} c_m^* \cdot c_n \cdot \underbrace{e^{-i(E_n - E_m)t/\hbar}}_{\delta_{n,m} \text{ im Mittel}} \cdot \langle \psi_n(x) | \psi_m(x) \rangle \right\rangle_t = \sum_n |c_n|^2 \cdot |\psi_n(x)|^2$$

Im Falle des asymmetrischen Doppelmuldenpotentials wissen wir, dass die Eigenzustände zu niedrigen Eigenenergien *entweder links oder rechts* liegen. Liegt das Wellenpaket am Anfang in der linken Mulde, so regt $\varphi_0(x)$ nur die linken Eigenfunktionen stark an und eine Tunnelung in die rechte Mulde ist ziemlich unwahrscheinlich.

Bei der symmetrischen Doppelmulde ist dies anders: Die niedrigen Eigenfunktion liegen *sowohl links als auch rechts*. Das Anfangswellenpaket regt also beide Mulden an, und liegt folglich im Mittel in beiden Mulden. In der Zeitentwicklung erkennt man ein periodisches Wechseln der Wahrscheinlichkeiten zwischen beiden Mulden.

5.4 Phasenraumdarstellung

Wir möchten die Quantenmechanik (Eigenfunktionen und Wellenpaketdynamik) mit klassischer Phasenraumdynamik (Ort und Impuls eines Teilchens) vergleichen. Wir müssen also Ort und Impuls eines Wellenpaketes bestimmen. Allerdings liegen Zustände in der Quantenmechanik *entweder* in Orts- *oder* in Impulsdarstellung vor. Stattdessen werden wir das Wellenpaket auf definierte Zustände im Phasenraum projizieren (**Husimi-Darstellung**).

5.4.1 Kohärenter Zustand

Ein Gaußsches Wellenpaket am Phasenraumpunkt (p_0, x_0) bezeichnet man als **kohärenten Zustand** $|\alpha(p_0, x_0)\rangle$. Seine Ortsdarstellung ist:

$$\alpha(x, p_0, x_0) = \langle x | \alpha(p_0, x_0) \rangle = \sqrt[4]{\frac{1}{2\pi \cdot (\Delta x)^2}} \cdot \exp \left[-\frac{(x - x_0)^2}{4 \cdot (\Delta x)^2} + \frac{i}{\hbar} \cdot p_0 \cdot x \right]$$

Dieses Wellenpaket ist normiert. Seine Impulsdarstellung hat fast dieselbe Gauß-Glockenstruktur wie die Ortsdarstellung:

$$\begin{aligned}\tilde{\alpha}(p, p_0, x_0) &= \langle p | \alpha(p_0, x_0) \rangle = \int_{-\infty}^{\infty} \langle p | x \rangle \cdot \langle x | \alpha(p_0, x_0) \rangle dx \\ &= \sqrt{\frac{1}{2\pi \cdot (\Delta p)^2}} \cdot \exp \left[-\frac{(p-p_0)^2}{4 \cdot (\Delta p)^2} - \frac{i}{\hbar} \cdot x_0 \cdot (p - p_0) \right] \\ \text{mit } \Delta p &= \frac{\hbar}{2\Delta x}\end{aligned}$$

Die Unschärfen des Wellenpaketes sind minimal ($\Delta x \cdot \Delta p = \hbar/2$ bzw. $\Delta x \cdot \Delta p = \hbar_{\text{eff}}/2$ im einheitenlosen Fall). Es ist entweder Δx oder Δp frei wählbar. Kohärente Zustände bilden ein Erzeugendensystem, aber *keine* Basis des Hilbertraumes, da sie nicht orthogonal sind.

Der Freiheitsgrad in der Wahl der Unschärfen wird meist durch einen Asymmetrieparameter A beschrieben und damit vom (effektiven) Planckschen Wirkungsquantum unabhängig:

$$\Delta x = \sqrt{\frac{\hbar_{\text{eff}}}{2}} \cdot A \quad \Rightarrow \quad \Delta p = \sqrt{\frac{\hbar_{\text{eff}}}{2}} \cdot \frac{1}{A}$$

Die physikalische Fragestellung in diesem Verfahren besteht also in der sinnvollen Wahl von A , welches stark von der Struktur der klassischen Phasenraumtrajektorien abhängt.

5.4.2 Husimi-Darstellung

Die Projektion eines Zustandes $|\varphi\rangle$ auf kohärente Zustände bei (p_0, x_0) liefert die **Husimi-Funktion**:

$$H_{\varphi}(p_0, x_0) = \frac{1}{\hbar} \cdot |\langle \alpha(p_0, x_0) | \psi \rangle|^2 = \frac{1}{\hbar} \cdot \left| \int_{-\infty}^{+\infty} \alpha^*(x, p_0, x_0) \cdot \varphi(x) dx \right|^2$$

Die Husimi-Funktion ist positiv und normiert ($\iint_{-\infty}^{+\infty} H_{\varphi}(p_0, x_0) dp_0 dx_0 = 1$) und kann als Wahrscheinlichkeitsdichte interpretiert werden. Eine Rücktransformation auf den Zustand ist aber nicht möglich, wie die folgende Projektion auf Ort und Impuls zeigt:

$$\begin{aligned}\int_{-\infty}^{+\infty} H_{\varphi}(p_0, x_0) dp_0 &= \int_{-\infty}^{+\infty} |\varphi(x)|^2 \cdot \frac{1}{\sqrt{2\pi \cdot (\Delta x)^2}} \cdot \exp \left[-\frac{(x-x_0)^2}{2 \cdot (\Delta x)^2} \right] dx \\ \int_{-\infty}^{+\infty} H_{\varphi}(p_0, x_0) dx_0 &= \int_{-\infty}^{+\infty} |\tilde{\varphi}(p)|^2 \cdot \frac{1}{\sqrt{2\pi \cdot (\Delta p)^2}} \cdot \exp \left[-\frac{(p-p_0)^2}{2 \cdot (\Delta p)^2} \right] dp\end{aligned}$$

In beiden Fällen ist also der Zustand mit einem Gauß-Wellenpaket verfaltet.

Um eine geeignete Basisfunktion im Phasenraum zu wählen, vergleicht man die Basisfunktionen im Phasenraum mit dem klassischen Phasenraum. Wird der Bereich niedriger Energien gut abgedeckt, so reichen auch wenige Basisfunktionen.

5.4.3 Semiklassik

Wir wollen nun den halbklassischen Grenzfall $\hbar_{\text{eff}} \rightarrow 0$ diskutieren. Die Basisfunktionen leben dann in einem kleineren Phasenraumbereich und die Eigenfunktionen nähern sich immer mehr den klassischen Bahnen, also den Konturlinien von $H(p, x)$, an. Ihre Breite fällt $\sim \sqrt{\hbar_{\text{eff}}}$ ab. Für mehrdimensionale Probleme wird die semiklassische Dynamik viel komplizierter (**Quantenchaos**).

5.5 Räumlich periodische Systeme

5.5.1 Bloch-Theorem

Die Standardsituation in der Festkörperphysik ist das Vorliegen eines räumlich periodischen Potentials. Im eindimensionalen Fall, den wir hier behandeln, ist also:

$$H = \frac{p^2}{2m} + V(x) \quad \text{mit} \quad V(x+a) = V(x)$$

Die Eigenfunktionen $\psi(x)$ des Hamilton-Operators sind fast periodisch. Das **Bloch-Theorem** besagt nämlich, dass alle Eigenfunktionen die folgende Struktur haben:

$$\begin{aligned} \psi_k(x) &= e^{ikx} \cdot u_k(x) \quad \text{mit} \quad u_k(x+a) = u_k(x) \\ \psi_k(x+a) &= e^{ik(x+a)} \cdot u_k(x) = e^{ika} \cdot \psi_k(x) \end{aligned}$$

Die Aufenthaltswahrscheinlichkeit $|\psi_k(x)|^2 = |u_k(x)|^2$ der Eigenfunktionen ist also periodisch, die Amplitude jedoch nicht! Man kann nun die Eigenfunktionen $\psi_k(x)$ zu k und $k + 2\pi/a$ identisch wählen (Reduzierung des Problems auf die erste Brillouin-Zone):

$$u_{k+\frac{2\pi}{a}}(x) = e^{-i \cdot \frac{2\pi}{a}} \cdot u_k(x)$$

Dann ist $k \in]-\frac{\pi}{a}, \frac{\pi}{a}]$ der **Bloch-Wellenvektor** (im eindimensionalen Fall auch als Bloch-Phase bezeichnet). Jedes k liefert einen anderen Satz von Eigenfunktionen $\psi_{k,n}(x)$.

5.5.2 Bandspektrum

Nun sei das Potential ähnlich wie im Festkörper eine periodische Ansammlung von Mulden.

Im Fall niedriger Energien treten schmale Bänder um die Eigenenergien E_i einzelner Muldenpotentiale auf. Das periodische System kann mit der **Tight-Binding-Näherung** beschrieben werden, nach der neben der aktuellen Mulde nur das Tunneln zu direkten Nachbarmulden noch zusätzlich in die Rechnung eingeht.

$$H_\infty = \begin{pmatrix} \ddots & \ddots & & \dots & 0 \\ \ddots & E_i & t_i & & \vdots \\ & t_i & E_i & t_i & \\ & & t_i & E_i & t_i \\ \vdots & & & t_i & E_i & \ddots \\ 0 & \dots & & & \ddots & \ddots \end{pmatrix}$$

Numerische Umsetzung

Zunächst muss das System wieder diskretisiert werden; zusätzlich reduziert man es auf eine Einheitszelle. Das Bloch-Theorem liefert eine Verbindung zwischen den Zellen: $\psi_k(x+1) = e^{ik} \cdot \psi_k(x)$ und damit:

$$\psi_N = e^{ik} \cdot \psi_0 \quad \text{bis} \quad \psi_{-1} = e^{-ik} \cdot \psi_{N-1}$$

6 Fourier-Transformation

Wir betrachten zunächst die kontinuierliche Fourier-Transformation (FT):

$$F(\nu) = \int_{-\infty}^{\infty} f(t) \cdot e^{-2\pi i \cdot \nu t} dt \quad \Leftrightarrow \quad f(t) = \int_{-\infty}^{\infty} F(\nu) \cdot e^{2\pi i \cdot \nu t} d\nu$$

Die Fourier-Transformierte beantwortet die Frage, welche Frequenzen in einem Signal enthalten sind, zum Beispiel:

$$f(t) = \sin(2\pi t) \quad \Rightarrow \quad F(\nu) = \frac{1}{2i} \cdot [\delta(\nu - 1) - \delta(\nu + 1)]$$

Man beachte, dass in den Definitionsgleichungen für F und f keine Vorfaktoren der Form $1/2\pi$ oder $1/\sqrt{2\pi}$ vorkommen, da wir nach der Frequenz, nicht nach der Kreisfrequenz transformieren.

6.1 Fourier-Reihen

Für periodische Funktionen der Periode T , also f mit $f(t + T) = f(t)$, sind die Frequenzen diskret:

$$c_n = \int_0^T f(t) \cdot e^{-2\pi i \cdot nt/T} dt \quad \Rightarrow \quad f(t) = \frac{1}{T} \cdot \sum_{n \in \mathbb{Z}} c_n \cdot e^{2\pi i \cdot nt/T}$$

Hierbei sind c_n die **Fourier-Koeffizienten** der Fourier-Reihe, zum Beispiel:

$$f(t) = \sin(2\pi t) \quad \Rightarrow \quad c_n = \begin{cases} \pm \frac{1}{2i} & n = \pm 1 \\ 0 & \text{sonst} \end{cases}$$

Das Ziel ist die approximative Rekonstruktion von $f(t)$ durch Fourier-Summen:

$$f_N(t) = \frac{1}{T} \sum_{\substack{n \in \mathbb{Z} \\ |n| \leq N}} c_n \cdot e^{2\pi i \cdot nt/T}$$

Hiernach ist $f_N(t)$ eine Näherung für $f(t)$ und hat eine Periode T . Für $N \rightarrow \infty$ konvergiert f_N , nach dem Satz von Dirichlet, im Allgemeinen punktweise gegen f , aber nicht gleichmäßig. Zum Beispiel sieht man bei einer punktuell unstetigen Funktion in der neunten Übung ein Überschießen von 18 % („Gibbs' Phänomen“). Desweiteren entspricht c_0 dem zeitlichen Mittelwert von f , auch weitere Eigenschaften der Funktion übertragen sich auf die Koeffizienten:

$$\begin{aligned} f \text{ reell:} \quad c_{-n} &= c_n^* \\ f \text{ imaginär:} \quad c_{-n} &= -c_n^* \\ f \text{ gerade:} \quad c_{-n} &= c_n \\ f \text{ ungerade:} \quad c_{-n} &= -c_n \end{aligned}$$

Bei einer analytischen Funktion fallen die Koeffizienten schneller ab als jede Potenz von $\frac{1}{n}$. Enthält die Funktion aber Sprungstellen, so gehen die Koeffizienten mit $\frac{1}{n}$.

6.2 Abtasttheorem

Wir tasten eine Funktion $f: \mathbb{R} \rightarrow \mathbb{R}$ in diskreten Schritten ab:

$$f(t) \rightarrow f_k = f(k \cdot \Delta) \Rightarrow t_k = k \cdot \Delta$$

Relevante Parameter sind hierbei die Abtastrate $\frac{1}{\Delta}$ und die **Nyquist-Frequenz** $\nu_c := 1/2\Delta$. Es gilt nun herauszufinden, wie fein Δ für den jeweiligen Zweck gewählt werden muss.

Sei $f(t)$ eine durch ν_{\max} bandbegrenzte Funktion, also:

$$f(t) = \int_{-\nu_{\max}}^{\nu_{\max}} F(\nu) \cdot e^{2\pi i \cdot \nu t} d\nu$$

Das **Abtasttheorem** besagt: Wenn die Abtastrate so groß ist, dass $\nu_c \geq \nu_{\max}$ wird (also $\Delta < 1/2\nu_{\max}$), dann ist $f(t)$ exakt aus den f_k rekonstruierbar:

$$f(t) = \sum_{k=-\infty}^{\infty} f_k \frac{\sin(2\pi i \cdot \nu_c \cdot (t - t_k))}{2\pi \cdot \nu_c \cdot (t - t_k)}$$

Somit reichen zwei Punkte pro Periode. Zum Beispiel ist für die Audiodaten einer CD die Nyquist-Frequenz $\nu_c = 22\,050$ Hz und die Abtastrate $1/\Delta = 44\,100$ Hz. Das größere Problem ist das Aliasing für nicht-bandbegrenzte $f(t)$. Eine diskrete Abtastung bei falschen Frequenzen würde dann zu einem mitunter signifikanten Informationsverlust führen (siehe Skizze rechts), da Beiträge $|\nu| > \nu_{\max}$ auf den Frequenzbereich $[-\nu_{\max}, \nu_{\max}]$ abgebildet würden. Dies umgeht man in der Praxis, indem man das Signal zunächst mit einem Bandfilter auf den gewünschten Frequenzbereich reduziert.

GRAPHIK

(wird nachgereicht)

Informationsverlust beim Sampling

6.3 Diskrete Fourier-Transformation

Für eine auf $[0, T)$ gegebene periodische Funktion f sei die diskrete Abtastung mit einer geraden Anzahl N von Zwischenpunkten auf diesem Intervall gegeben:

$$t_k = k \cdot \Delta \quad \text{mit} \quad \Delta = \frac{T}{N} \quad \text{und} \quad f_k = f(t_k) \quad \text{für} \quad k = 0, 1, \dots, N-1$$

Das Ziel ist die Schätzung der Fourier-Transformation für die N Frequenzen $\nu_n = n/(N \cdot \Delta)$.

$$c_n \equiv F(\nu_n) = \int_0^T f(t) \cdot e^{-2\pi i \cdot \nu_n t} dt \approx \sum_{k=0}^{N-1} f_k \cdot e^{-2\pi i \cdot \nu_n t_k} \cdot \Delta = \Delta \cdot F_n$$

Die diskrete Fouriertransformation sieht also wie folgt aus:

$$F_n := \sum_{k=0}^{N-1} f_k \cdot e^{-2\pi i \cdot k_n/N} \Rightarrow f_k = \frac{1}{N} \cdot \sum_{n=0}^{N-1} F_n \cdot e^{2\pi i \cdot k_n/N}$$

Diese Gleichungen sind dimensionslos. Wegen der Periodizität $F_{n+N} = F_n$ interpretiert man die Koeffizienten $F_{N/2}$ bis F_{N-1} üblicherweise als $F_{-N/2}$ bis F_{-1} . Für die Numerik gilt $f_{-k} = \text{fft}^{-1}(\text{fft}(f_{-k}))$ nur im Rahmen der numerischen Genauigkeit.

6.4 Schnelle Fourier-Transformation (FFT)

Ursprünglich haben wir folgendes als Fouriertransformation:

$$F_n = \sum_{k=0}^{N-1} W^{nk} \cdot f_k \quad \text{mit} \quad W = e^{-2\pi i/N} \quad n = 0, 1, \dots, N-1$$

Zu deren Ausführung sind N^2 Operationen nötig, hingegen benötigt die FFT für $N = 2^l$ nur $N \cdot \log_2 N$ Operationen. Die Idee besteht darin, eine Fouriertransformation mit N Punkten in zwei Fouriertransformationen mit je $N/2$ Punkten aufzuteilen:

$$F_n = \sum_{j=0}^{N/2-1} W^{n \cdot 2j} \cdot f_{2j} + \sum_{j=0}^{N/2-1} W^{n \cdot (2j+1)} \cdot f_{2j+1} \hat{=} F_n^{\text{gerade}} + W^n \cdot F_n^{\text{ungerade}}$$

Ist N keine Potenz von 2, so muss man die Punktmenge anders zerlegen, idealerweise durch eine Primfaktorzerlegung.

7 Solitonen in der Physik

Solitonen sind bestimmte Lösungen nichtlinearer Wellengleichungen. 1845 beschrieb J. S. Russell die Ausbreitung einer Wasserwelle vom 10 m Länge und 0,5 m Höhe, deren Form sich nur wenig änderte (keine Dispersion) mit einer Ausbreitungsgeschwindigkeit $c = \sqrt{g(h+a)}$ (mit h – Tiefe des Wassers, a – Amplitude). Er bezeichnete dieses Phänomen als „solitary wave“ („einsame Welle“).

7.1 Korteweg-de-Vries-Gleichung

Die folgende Standardform der Korteweg-de-Vries-Gleichung wurde zuerst 1894 angegeben. Sie beschreibt die Propagation eines Soliton-Wellenpaketes $u(x, t)$:

$$\frac{\partial u}{\partial t} + 6 \cdot u \cdot \frac{\partial u}{\partial x} + \frac{\partial^3 u}{\partial x^3} = 0$$

Aufgrund des zweiten Termes ist diese Gleichung *nichtlinear*. Außerdem ist auffällig, dass es sich um eine partielle Differentialgleichung *dritter Ordnung*. Das Hauptproblem beim Lösen dieser Gleichung ist, dass das lineare Superpositionsprinzip nicht angewendet werden kann.

Zur Lösung verwenden wir den Ansatz $u(x, t) = f(\xi)$ mit $\xi = x - ct$ und erhalten:

$$-c \cdot f' + \underbrace{6 \cdot f \cdot f'}_{3 \cdot (f^2)'} + f''' = 0$$

Einmalige Integration liefert: (unter Hinzugewinnung einer Integrationskonstante $A \in \mathbb{R}$)

$$-c \cdot f + 3 \cdot f^2 + f'' = A$$

Wir fügen f' als integrierenden Faktor hinzu:

$$\begin{aligned} -c \cdot f \cdot f' + 3 \cdot f' \cdot f^2 + f' \cdot f'' &= A \cdot f' \\ -\frac{c}{2} \cdot (f^2)' + (f^3)' + \frac{1}{2} \cdot [(f')^2]' &= A \cdot f' \end{aligned}$$

Nun können wir nochmals integrieren und erhalten eine weitere Integrationskonstante $B \in \mathbb{R}$:

$$-\frac{c}{2} \cdot f^2 + f^3 + \frac{1}{2} \cdot (f')^2 = A \cdot f + B$$

Als Randbedingungen wählen wir $f, f', f'' \rightarrow 0$ für $\xi \rightarrow \pm\infty$. Somit muss $A = B = 0$ sein:

$$(f')^2 + f^2 \cdot (2f - c) = 0$$

Diese Gleichung kann man durch einfache Separation der Variablen lösen:

$$\int \frac{df}{f \cdot \sqrt{2f - c}} = \pm \int d\xi$$

Zur Lösung des linken Integrals bemühen wir die Substitution $f = \frac{c}{2} \cdot \text{sech}^2 \Theta$ mit dem Sekans hyperbolicus $\text{sech}(x) = 1/\cosh(x)$. Schließlich gelangt man auf:

$$u(x, t) = f(x - ct) = \frac{c}{2} \cdot \text{sech}^2 \left[\frac{\sqrt{c}}{2} \cdot (x - ct) \right]$$

Die Lösungen haben folgende Eigenschaften:

- Sie sind lokalisiert. (Der Sekans hyperbolicus ist in erster Näherung wie eine Gaussglocke geformt.)
- Die Amplitude $c/2$ ist die Hälfte der Geschwindigkeit c .
- Die Breite ist proportional zu c^{-1} .

Obwohl die Gleichung nichtlinear ist, gibt es also stabile Lösungen! Dies liegt an dem Zusammenspiel der einzelnen Terme in der Korteweg-de-Vries-Gleichung, wie man sieht, wenn man einzelne Terme weglässt:

- Beim Weglassen des nichtlinearen Termes entsteht Dispersion:

$$u_t + u_{xxx} = 0 \quad \Rightarrow \quad u(x, t) = e^{i(kx + \omega t)} \quad \text{mit} \quad \omega = k^3 \quad \Rightarrow \quad c = \frac{\omega}{k} = k^2$$

Der Term u_{xxx} wirkt also stark dissipativ, denn alleine verursacht er eine starke Dispersion.

- Betrachtet man nur den nichtlinearen Term, so ergibt sich:

$$u_t + u \cdot u_x = 0 \quad \Rightarrow \quad u(x, t) = f\left(x - u(x, t) \cdot t\right)$$

Die Bestimmung von $u(x, t)$ aus einem gegebenen $u(x, 0) = f(x)$ ist ein nicht-triviales Problem.

Das Wechselspiel dieser beiden Teile der Gleichung führt zur Stabilität der Solitonen.

7.2 Numerik

Zur numerischen Lösung der Korteweg-de-Vries-Gleichung setzen wir die **Pseudospektralmethode** nach Formberg, Whitham ein.

$$\begin{aligned} u_x(x, t) &= \frac{\partial u(x, t)}{\partial x} = i \cdot [\mathcal{F}^{-1} \alpha](x, t) \quad \text{mit} \quad \alpha(k, t) = k \cdot \tilde{u}(k, t) = k \cdot [\mathcal{F} u](k, t) \\ u_{xxx}(x, t) &= \frac{\partial^3 u(x, t)}{\partial x^3} = -i \cdot [\mathcal{F}^{-1} \beta](x, t) \quad \text{mit} \quad \beta(k, t) = k^3 \cdot \tilde{u}(k, t) = k^3 \cdot [\mathcal{F} u](k, t) \end{aligned}$$

Durch \mathcal{F} bzw. \mathcal{F}^{-1} seien die Fourier-Transformation sowie -Rücktransformation bezeichnet:

$$\begin{aligned} \tilde{f}(k) &= [\mathcal{F} f](k) = \frac{1}{\sqrt{2\pi}} \cdot \int_{-\infty}^{\infty} f(x) \cdot e^{-ikx} dx \\ f(x) &= [\mathcal{F}^{-1} \tilde{f}](x) = \frac{1}{\sqrt{2\pi}} \cdot \int_{-\infty}^{\infty} \tilde{f}(k) \cdot e^{ikx} dx \end{aligned}$$

Nun fehlt noch die Diskretisierung der Zeitableitung:

$$u(x, t + \Delta t) = u(x, t - \Delta t) - 2 \cdot \Delta t \cdot [6 \cdot u(x, t) \cdot u_x(x, t) + u_{xxx}(x, t)]$$

Hierin setzen wir die obigen Darstellungen der Ortsableitungen ein:

$$u(x, t + \Delta t) = u(x, t - \Delta t) - 2i \cdot \Delta t \cdot [6 \cdot u(x, t) \cdot [\mathcal{F}^{-1} \alpha](x, t) - [\mathcal{F}^{-1} \beta](x, t)]$$

Nun betrachten wir anstelle von β die Funktion $\gamma(k, t) = \sin(\Delta t \cdot k^{-1}) \cdot \tilde{u}(k, t)$, also:

$$u(x, t + \Delta t) = u(x, t - \Delta t) - 2i \cdot \left[6\Delta t \cdot u(x, t) \cdot [\mathcal{F}^{-1}\alpha](x, t) - [\mathcal{F}^{-1}\gamma](x, t) \right]$$

Dies ist besser, weil Störungen bei großem k gut gedämpft werden.¹

Numerische Implementation

Wir betrachten die Funktion auf einem diskretisierten Ortsintervall:

$$x \in [-L, L] \quad \text{mit} \quad x_j = j \cdot \Delta x - L \quad \text{für} \quad j = 0, 1, \dots, N-1 \quad \text{mit} \quad \Delta x = \frac{2 \cdot L}{N}$$

Für die Wahl der Zeitschritte nutzen wir eine Stabilitätsbedingung:

$$\Delta t \leq \frac{3 \cdot (\Delta x)^3}{2\pi^2}$$

Beim ersten Zeitschritt funktioniert das obige Zeitdiskretisierungsschema nicht, da neben dem Anfangsbedingungen ein weiterer Schritt benötigt würde, daher verwenden wir als ersten Integrationsschritt einen Euler-Schritt:

$$u(x, t + \Delta t) = u(x, t) - 1i \cdot \left[6\Delta t \cdot u(x, t) \cdot [\mathcal{F}^{-1}\alpha](x, t) - [\mathcal{F}^{-1}\gamma](x, t) \right]$$

Diese Numerik benötigt keine extra Routine. Zur Berechnung von $[\mathcal{F}^{-1}\alpha](x, t)$ ermitteln wir als erstes $\alpha(k, t) = k \cdot [\mathcal{F}u](k, t)$ mit $K = k_l = k_0 + l \cdot \Delta k$:

$$\begin{aligned} k_l \cdot [\mathcal{F}u](k_l) &= k_l \cdot \frac{1}{\sqrt{2\pi}} \cdot \int_{-\infty}^{\infty} e^{-i \cdot k_l x} \cdot u(x) \, dx \\ &\approx k_l \cdot \frac{1}{\sqrt{2\pi}} \cdot \int_{-L}^L e^{-i \cdot k_l x} u(x) \, dx \\ &\approx k_l \cdot \frac{\Delta x}{\sqrt{2\pi}} \cdot \sum_{j=0}^{N-1} e^{-i \cdot k_l \cdot (x_0 + j \cdot \Delta x)} \cdot u(x_j) \end{aligned}$$

Hieraus bestimmen wir nun Δk :

$$e^{-i \cdot k_l \cdot (x_0 + j \cdot \Delta x)} \stackrel{!}{=} e^{-i \cdot (k_0 + l \cdot \Delta k) \cdot (x_0 + j \cdot \Delta x)} = e^{-i \cdot k_0 x_0} \cdot e^{-i \cdot j \cdot k_0 \Delta x} \cdot e^{-i \cdot l \cdot \Delta k x_0} \cdot e^{-i \cdot l \cdot j \cdot \Delta k \Delta x}$$

Daraus folgt $\Delta k = \frac{2\pi}{N \cdot \Delta x} = \frac{\pi}{L}$, somit ist die Form passend für FFT.

Nun führen wir die Rücktransformation $[\mathcal{F}^{-1}\alpha](x, t)$ aus:

$$\begin{aligned} [\mathcal{F}^{-1}\alpha](x_m) &= \frac{1}{\sqrt{2\pi}} \cdot \int_{-\infty}^{\infty} e^{i \cdot k x_m} \cdot \alpha(k) \, dk \\ &\approx \frac{1}{\sqrt{2\pi}} \cdot \int_{-k_{\max}}^{k_{\max}} e^{i \cdot k x_m} \cdot \alpha(k) \, dk \\ &\approx \frac{\Delta k}{\sqrt{2\pi}} \cdot \sum_{l=0}^{N-1} e^{i \cdot k_l x_m} \cdot \alpha(k_l) \quad \Rightarrow \quad k_0 = -\frac{N}{2} \Delta k = -\frac{\pi \cdot N}{2L} \end{aligned}$$

¹Es bietet sich an, dies bei Bearbeitung der Übung zu prüfen.

Einsetzen des Ausdruckes für α liefert:

$$[\mathcal{F}^{-1}\alpha](x_m) = \frac{\Delta k \cdot \Delta x}{2\pi} \cdot \sum_{l=0}^{N-1} e^{i \cdot k_l x_m} \cdot e^{-i \cdot k_0 x_0} \cdot e^{-i \Delta k \cdot x_0 \cdot l} \cdot k_l \cdot \sum_{j=0}^{N-1} e^{-i l j \cdot \Delta k \Delta x} \cdot e^{-i j \cdot k_0 \Delta x} \cdot u(x_j)$$

Unter Beachtung der Beziehung

$$e^{i \cdot k_l x_m} = e^{i \cdot k_0 x_0} \cdot e^{i m \cdot k_0 \Delta x} \cdot e^{i l \cdot \Delta k x_0} \cdot e^{i l m \cdot \Delta k \Delta x}$$

erhält man den wesentlich einfacheren Ausdruck:

$$\begin{aligned} [\mathcal{F}^{-1}\alpha](x, t) &\approx \underbrace{\frac{\Delta x \Delta k}{2\pi}}_{=1/N} \cdot e^{i \cdot k_0 \Delta m} \cdot \sum_{l=0}^{N-1} e^{i \Delta k \Delta x \cdot l \cdot m} \cdot k_l \cdot \sum_{j=0}^{N-1} e^{-1 \Delta x \Delta k \cdot l \cdot j} \cdot e^{-i k_0 \Delta x \cdot j} \cdot u(x_j) \\ &= e^{-i \pi \cdot m} \cdot \frac{1}{N} \cdot \sum_{l=0}^{N-1} e^{2\pi i \cdot \frac{lm}{N}} \cdot k_l \cdot \sum_{j=0}^{N-1} e^{-2\pi i \cdot \frac{l j}{N}} \cdot \underbrace{e^{i \pi \cdot j}}_{\equiv z_j} \cdot u(x_j) \\ &= \bar{z} \cdot \text{fft}[k \cdot \text{fft}(z \cdot u)] \end{aligned}$$

Hierbei sind `fft` und `ifft` die diskreten Fourier-Transformationen von SciPy.

Stichwortverzeichnis

A

Abtasttheorem	30
Aliasing	30

B

Bloch-Phase	28
Bloch-Theorem	28
Bloch-Wellenvektor	28

D

Differentialgleichung	14
gewöhnliche	14
partielle	14
steife	17
Drei-Achtel-Regel	12

F

Fehler	
globale	15
lokaler	15
Fourier-Koeffizienten	29
Fourier-Reihe	29
Fourier-Transformation	
kontinuierliche	29

G

Gauß-Methode	12
gekickter Rotor	6
Gleichung	
einheitenlose	6

H

Hermite-Polynom	22
Husimi-Darstellung	26
Husimi-Funktion	27

K

kohärenter Zustand	26
--------------------------	----

M

Milne-Regel	12
Mittelpunktregel	11

N

Newton-Cotes-Formeln	10
Nyquist-Frequenz	30

P

Pseudospektralmethode	33
-----------------------------	----

Q

Quantenchaos	27
--------------------	----

R

Romberg-Methode	12
-----------------------	----

S

Simpson-Regel	11
Soliton	32
Superpositionsprinzip	20

T

Tight-Binding-Näherung	28
Trapezregel	11

V

Variable	
einheitenlose	6

W

Weddie's Regel	12
----------------------	----

Z

Zahlen	
Festkomma-	7
ganze	7
Gleitkomma-	7
Normalisierung	7