

chem1004D:

Praktikum

Computerchemie/Theoretische Chemie:

I hear and I forget.
I see and I remember.
I do and I understand.

(chinesisches Sprichwort)

Ziele

- konkreter Einblick in Algorithmen/Programme zur Lösung der wesentlichen Aufgaben der Theoretischen Chemie (stationäre Quantenzustände, klassische und quantenmechanische Dynamik)
- durch eigenen Umgang mit diesen Programmen Erfahrung mit deren Verhalten und Ergebnissen:
 - zu erwartende Genauigkeit/Fehler
 - Wichtigkeit der korrekten Wahl von Basisfunktionen und Parametern
 - Größenskalierung der Algorithmen
- dadurch besseres Verständnis vorgefertigter Anwendungspakete.

Themenübersicht

- 1: Mini-Einführung: Fortran, gnuplot
- 2: Numerik: Integration, Matrixdiagonalisierung, Differentialgleichungen
- 3: klassisch-mechanische Dynamik: einfacher Verlet-Algorithmus; harmonischer Oszillator und Morse-Oszillator; Übertragbarkeit auf größere chemische Systeme
- 4: Lösung der zeitunabhängigen Schrödingergleichung durch Basisentwicklung:
 - die dabei zu lösenden Teilaufgaben und deren Charakteristika: analytische oder numerische Integration → Matrixelemente, Matrixdiagonalisierung → Eigenwerte und Eigenvektoren
 - Basis: Sinusfunktionen; praktische Aspekte der korrekten Wahl von Basisfunktionen
 - Übertragbarkeit auf größere chemische Systeme; Gemeinsamkeiten mit (und Unterschiede zu) Verfahren der Quantenchemie
- 5: Zeitabhängige Wellenfunktionen durch Superposition von Eigenzuständen: die Welt jenseits gequantelter stationärer Zustände
- 6: quantenmechanische Dynamik:
 - Gauß-Wellenpaket im Harmonischen Oszillator und Morse-Oszillator; Vergleich zur klassischen Mechanik
 - Berechnung von Spektren durch Fouriertransformation der Autokorrelationsfunktion

1 Kurzeinführung Fortran

Siehe separates „Fortran“-Skript!

Kurzeinführung gfortran-Compiler

Kurzeinführung ddd-Debugger

Kurzeinführung Zeichenprogramm gnuplot

2 Numerik

2.1 Numerische Integration

Vorbemerkungen:

Wann brauchen wir numerische Integration?

- *nicht* bei analytisch integrierbaren Funktionen, *sondern*
- bei analytisch gegebenen, aber nicht analytisch integrierbaren Funktionen,
- bei nur numerisch gegebenen Funktionen.

Einzigste Voraussetzung an zu integrierende Funktion $f(x)$:

$f(x)$ muß berechenbar sein, für einige diskrete, vorgegebene Werte von x innerhalb des Integrationsintervalls $[a, b]$.

Grundprinzip:

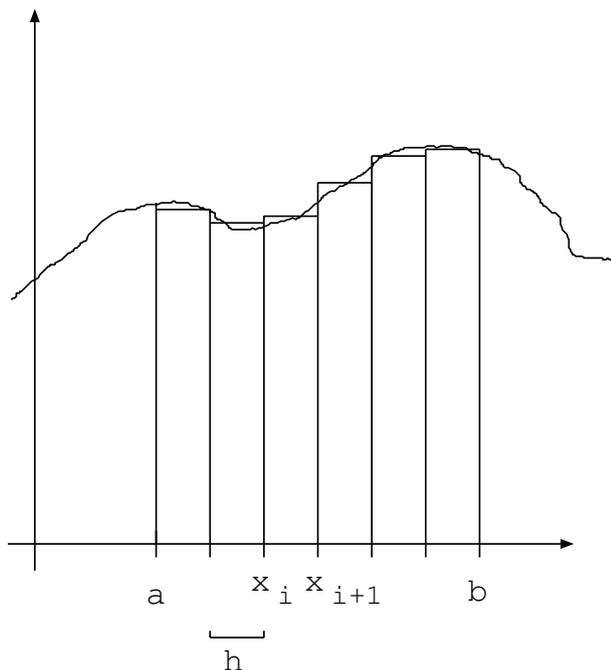
$$\int_a^b f(x) dx = \sum_{i=1}^N c_i f(x_i) + \text{Fehler}(N, \dots) \quad (1)$$

- unabhängig von der Form von $f()$,
- Lage der Stützstellen x_i bestimmt Werte der Koeffizienten c_i ,
- N möglichst klein,
- Fehler für gegebenes N möglichst klein und abschätzbar,
- Fehler soll für wachsendes N schnell gegen Null gehen.

Alle Integrationsverfahren dieses Typs machen implizite Annahmen über den Verlauf von $f(x)$ *zwischen* den Integrationsstützstellen x_i . Das ist theoretisch und praktisch nicht so gefährlich wie es scheint.

Trapezregel (Euler-Verfahren)

(bei äquidistanten Stützstellen: Newton-Cotes-Verfahren)



$$\int_a^b f(x) dx \approx \langle f \rangle_a^b (b-a) \quad (2)$$

$$= h \sum_{i=1}^N \langle f \rangle_{x_i}^{x_{i+1}} \quad (3)$$

$$= h \sum_{i=1}^N \frac{1}{2} (f_i + f_{i+1}) \quad (4)$$

$$= h \left(\frac{1}{2} f_1 + f_2 + \dots + f_{N-1} + \frac{1}{2} f_N \right) \quad (5)$$

In jedem Intervall der Breite h wird $f(x)$ durch Gerade (Polynom 1.Ordnung) approximiert. Daraus folgt:

- die Formel ist exakt nur dann, wenn $f(x)$ eine Gerade ist;
- Wenn wir $f(x_{i+1})$ durch eine Taylorreihe um $f(x_i)$ approximieren:

$$f(x_{i+1}) = f(x_i) + hf'(x_i) + \frac{h^2}{2!} f''(x_i) + \dots \quad (6)$$

brechen wir hier also nach dem linearen Glied ab
 \Rightarrow der Fehler ist von der Ordnung

$$\mathcal{O}\left(\frac{1}{N^2}\right) = \mathcal{O}(h^2) \quad (7)$$

\Rightarrow doppelte Schrittanzahl bzw. halbe Schrittweite verringert den Fehler um einen Faktor 4.

Kontrolle der Konvergenz in der Praxis:

- einmalige Berechnung des Integrals (mit einem geratenen Wert für h) ist gefährlich; nur mit weiteren Informationen zu rechtfertigen (z.B. hinreichend genaue Resultate mit diesem h -Wert bei sehr ähnlichen Funktionen); sonst immer nötig;
- mehrmalige Berechnung des Integrals mit jeweils kleiner werdenden h -Werten, bis Änderung der Integralapproximation kleiner als vorgegebene Schranke ϵ .

Programmiertips zur Integration

Ein typisches Integrationsprogramm hat drei wesentliche Teile:

1. Funktion, die $f(x)$ für einen beliebigen input-Wert x berechnet;
2. Subroutine für die eigentliche Integration (bei festem N):
 - Parameter:
 - input: Integrationsgrenzen a, b
 - input: (feste) Anzahl von Stützstellen N
 - output: Integralwert s
 - berechne h aus a, b, N
 - setze $s = \frac{1}{2}f_1 + \frac{1}{2}f_N$
 - Schleife über alle $i = 1, N - 2$:
 - $s = s + f_i$ (wobei f_i ein Funktionsaufruf ist)
 - setze $s = s \times h$

3. Hauptprogramm

- user interface
- Schleife mit Zählvariable i :
 - berechne Integral s_i (Aufruf der subroutine)
 - Konvergenztest: ist $|s_i - s_{i-1}| < \epsilon$?
wenn ja, Ergebnisausgabe und Stop; sonst weiter
 - erhöhe N

Zum Testen Integral-Zwischenergebnisse s_i in jeder Iteration ausgeben lassen!

Fehleranfällige Stelle: Konvergenztest:

- Absolutwert der Differenz beachten!
- ϵ besser relativ nehmen, also testen, ob

$$|s_i - s_{i-1}| < \epsilon s_i \tag{8}$$

- die Computergenauigkeit ist begrenzt \Rightarrow man kann ϵ auch zu klein wählen!

Aufgabe zur numerischen Integration

Schreiben Sie in single-precision ein Integrationsprogramm auf der Basis der Trapezregel, für eine analytisch integrierbare Funktion Ihrer Wahl. Vergleichen Sie bei jeder Ausgabe das numerische Resultat mit dem analytischen.

Die folgenden Entwicklungsschritte sind zu empfehlen:

Stufe 1: keine Schleife über kleiner werdende Schrittweiten; stattdessen wird die Schrittweite interaktiv eingegeben und bleibt fest.

Stufe 2: in einer Schleife die Integration wiederholen und dabei die Anzahl der Gitterpunkte solange erhöhen, bis die Differenz der Ergebnisse zweier aufeinanderfolgender Integralberechnungen kleiner als ein am Anfang interaktiv einzugebender Toleranzwert ϵ wird.

Stufe 3: Experimentieren Sie mit verschiedenen ϵ -Werten und ermitteln Sie vernünftige Unter- und Obergrenzen für ϵ . Führen Sie dieselbe Untersuchung in double-precision aus und vergleichen Sie die Integrationsgenauigkeit und die ϵ -Werte mit dem single-precision-Fall.

2.2 Numerisches Eigenwertproblem (Matrixdiagonalisierung)

Für eine gegebene $n \times n$ -Matrix \mathbf{A} kann man nach

$$\mathbf{A}\vec{x} = \lambda\vec{x} \quad (9)$$

die zunächst unbekanntenen Eigenwerte λ und Eigenvektoren \vec{x} bestimmen.

Mit der Einheitsmatrix $\mathbf{1}$ läßt sich die Eigenwertgleichung Gl. (9) formal umformen zu:

$$(\mathbf{A} - \lambda\mathbf{1})\vec{x} = \tilde{\mathbf{A}}\vec{x} = \vec{0} \quad (10)$$

Dies ist ein *homogenes lineares Gleichungssystem* zur Bestimmung des unbekanntenen Vektors \vec{x} , bei dem allerdings die Matrix $\tilde{\mathbf{A}}$ noch nicht vollständig bekannt ist. Dennoch gilt, daß die Säkulardeterminante $\det(\tilde{\mathbf{A}}) = 0$ sein muß, damit es außer der uninteressanten trivialen Lösung $\vec{x} = \vec{0}$ noch weitere Lösungen gibt. Durch Auswertung der Determinante sieht man, daß die Forderung $\det(\tilde{\mathbf{A}}) = 0$ identisch zur Forderung ist, daß ein Polynom n -ter Ordnung in λ Null wird. Nach dem Fundamentalsatz der Algebra hat ein Polynom n -ter Ordnung genau n Nullstellen. Also hat eine (nicht-singuläre) $n \times n$ -Matrix immer genau n Eigenwerte (und dazugehörige Eigenvektoren); einige dieser Eigenwerte können allerdings identisch (entartet) sein.

Jede $n \times n$ -Matrix ist per Ähnlichkeitstransformation $\mathbf{T}^\dagger\mathbf{A}\mathbf{T}$ in diverse spezielle Formen überführbar, bei denen die Eigenwerte und Eigenvektoren der Matrix eine ausgezeichnete Rolle spielen (s. Stoer/Bulirsch, Kapitel 6). Zum Beispiel kann man jede $n \times n$ -Matrix unitär auf die Schur-Form transformieren (Dreiecksform, mit den Eigenwerten auf der Diagonalen). *Normale* Matrizen (mit $\mathbf{A}^\dagger\mathbf{A} = \mathbf{A}\mathbf{A}^\dagger$) sowie deren Unterklasse der *symmetrischen* (bzw. bei komplexen Zahlen: *hermiteschen*) Matrizen lassen sich immer unitär *diagonalisieren*:

$$\mathbf{U}^\dagger\mathbf{A}\mathbf{U} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix} \quad (11)$$

Die Transformationsmatrix \mathbf{U} besteht dabei aus den Eigenvektoren als Spaltenvektoren. Sie ist *orthogonal* (bzw. bei komplexen Zahlen: *unitär*), d.h. eine Transformation mit ihr läßt Längen und Winkel unverändert (= Drehungen und Spiegelungen im Anschauungsraum).

Eine Bestimmung der Nullstellen des aus der Säkulardeterminante resultierenden Polynoms in λ ist zwar prinzipiell möglich, aber numerisch ganz ungeeignet. Stattdessen konstruiert man z.B. im **Jacobi-Verfahren** iterativ die Transformationsmatrix \mathbf{U} , indem man die Außerdiagonalelemente von \mathbf{A} nacheinander auf Null transformiert. Dann enthält die Transformationsmatrix \mathbf{U} die Eigenvektoren als Spaltenvektoren und die resultierende Diagonalmatrix $\tilde{\mathbf{A}}$ die Eigenwerte auf der Diagonalen.

Verwendung von Bibliotheksroutinen zur Matrixdiagonalisierung

Die Matrixdiagonalisierung skaliert kubisch (n^3) mit der Matrixgröße n . Daher empfiehlt es sich in der Praxis, für die Diagonalisierung Profi-Programme aus Subroutinen-Bibliotheken zu verwenden, die hocheffizient arbeiten und besondere Matrixstrukturen ausnutzen können. Zum richtigen Umgang mit diesen Bibliotheken ist jedoch Hintergrundwissen in der Numerik notwendig.

Im vorliegenden Fall wird die LAPACK-Routine `dseyv` verwendet, die in doppelter Genauigkeit für eine reelle, symmetrische Matrix das Eigenwert-(eigenvalue)-Problem löst.

Falls das linken mit LAPACK Probleme bereiten sollte, werden die Routinen `jacobi` und `eigsrt` aus folgendem Buch zur Verfügung gestellt:

W. H. Press, B. P. Flannery, S. A. Teukolsky und W. T. Vetterling: „Numerical Recipes“, Cambridge University Press, Cambridge, 1990

`subroutine jacobi(a,n,lda,e,v,nrot):`
diagonalisiert die Matrix `a`

a: `real*8`-Matrix der Größe (n,n) ,
Eingabe: zu diagonalisierende Matrix (ist nachher zerstört)

n: `integer`,
Eingabe: tatsächliche Dimension des Diagonalisierungsproblems (hier: gleich Anzahl der Basisfunktionen) (Begrenzung $n < 100$ kann durch offensichtliche Modifikation des Programmtextes erhöht werden.)

lda: `integer`,
Eingabe: (feste) erste Dimension, mit der die Matrix `a` im aufrufenden Programm tatsächlich vereinbart wurde

e: `real*8`-Vektor der Länge (n) ,
Ausgabe: Eigenwerte (ungeordnet)

v: `real*8`-Matrix der Größe (n,n) ,
Ausgabe: enthält die Eigenvektoren als Spaltenvektoren (in derselben Ordnung wie im Vektor `e`)

nrot: `integer`,
Ausgabe: Anzahl der elementaren Iterationsschritte, die die Routine bis zur Lösung benötigt hat (hier weniger interessant).

`subroutine eigsrt(e,v,n,lda):`
Serviceprogramm zur Sortierung der Eigenwerte und -vektoren nach der Größe der Eigenwerte. Bedeutung der Parameter dieselbe wie bei `jacobi`.

Aufgabe zum Eigenwertproblem

Schreiben Sie ein Programm, das

- die Elemente einer $n \times n$ -Matrix (n vom Benutzer vorzugeben) mit Werten belegt (oder diese einliest),
- die Matrix zur Diagonalisierung an eine Bibliotheksroutine übergibt,
- und die Resultate (Eigenwerte und zugehörige Eigenvektoren) ausgibt.

Vorschläge zum Austesten:

(keine Panik in der Praxis: Das Vorzeichen eines Eigenvektors ist unbestimmt!)

- die Matrix $\begin{pmatrix} 5 & -2 \\ -2 & 2 \end{pmatrix}$,
Eigenwert: 1, Eigenvektor: $(1/\sqrt{5}, 2/\sqrt{5})^T$
Eigenwert: 6, Eigenvektor: $(-2/\sqrt{5}, 1/\sqrt{5})^T$

- die Matrix

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

mit den Eigenwerten $0, \sqrt{2}, -\sqrt{2}$ und den Eigenvektoren $\begin{pmatrix} a \\ 0 \\ -a \end{pmatrix}, \begin{pmatrix} a^2 \\ a \\ a^2 \end{pmatrix}, \begin{pmatrix} a^2 \\ -a \\ a^2 \end{pmatrix}$
mit $a = 1/\sqrt{2}$.

- Hückel-Matrix von Naphthalin:

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Eigenwerte: $\pm 0.618, \pm 1.0, \pm 1.303, \pm 1.618, \pm 2.303$

Versuchen Sie, die Eigenvektoren "chemisch" zu interpretieren.

2.3 Numerische Lösung von Differentialgleichungen

Vergleich zur Integration

Zwischen der Integration und der Lösung von Differentialgleichungen besteht ein enger Zusammenhang: Das Aufsuchen der Lösung $y(b)$ des Integrals

$$\int_a^b f(x) dx = y(b) \quad (12)$$

ist äquivalent zum Aufsuchen der Lösung von

$$\frac{dy}{dx} = f(x) \quad (13)$$

mit Anfangsbedingung $y(a) = 0$ und an einem Punkt $x = b$.

Differentialgleichungen vom Typ Gl. 13 können wir also mit den bekannten Integrationsverfahren bereits lösen. Die allgemeine Formel für Differentialgleichungen dieses Typs ist jedoch:

$$\frac{dy}{dx} = f(x, y) \quad (14)$$

Dabei ist y zunächst an (fast) keiner Stelle bekannt (bis auf Anfangs- oder Randbedingungen, s.u.), denn es ist ja die bei Differentialgleichungen gesuchte Größe. Alle Integrationsmethoden setzen jedoch voraus, daß wir $f()$ überall berechnen können, also für irgendein gegebenes x . Allerdings muß zumindest ein Punkt $y(x_0)$ als Zahl bekannt sein, da wir hier numerisch konkrete Zahlen berechnen und keine abstrakte Funktion $y(x)$. Also brauchen wir eine Art extrapolierendes Verfahren.

Außerdem gibt es bei Differentialgleichungen in der Praxis häufiger ein weniger gutartiges Verhalten von $y(x)$. Im Gegensatz zur Integration ist der Funktionsverlauf von $f()$ nicht von vornherein bekannt. Gute Differentialgleichungslöser müssen daher robuster und flexibler sein als die einfacheren Integrationsverfahren; zum Beispiel haben sie oft eine variable Schrittweite.

Euler-Cauchysches Polygonzugverfahren

Für eine Differentialgleichung

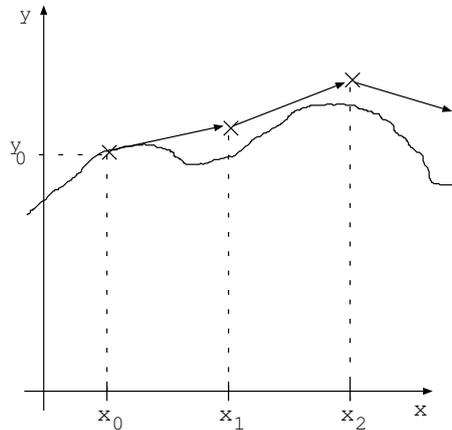
$$y' = \frac{dy}{dx} = f(x, y) \quad (15)$$

sei gegeben: ein funktionaler Ausdruck für $f(x, y)$ und eine Anfangsbedingung (x_0, y_0) . Gesucht sind weitere Punkte (x, y) der Funktion $y(x)$, die die Differentialgleichung Gl. 15 erfüllt.

Für *jeden* Punkt (x, y) gibt das bekannte $f(x, y)$ die *Steigung* der Funktion $y(x)$ an. Es liegt nahe, diese Information für eine linear-extrapolierende Konstruktion von $y(x)$ zu verwenden:

Wir machen also die Approximation

$$\frac{dy}{dx} \approx \frac{\Delta y}{\Delta x} \quad (16)$$



woraus sich ergibt

$$\Delta y = y_{i+1} - y_i = f(x_i, y_i) \Delta x \quad (17)$$

oder explizit, ausgehend vom Anfangswert (x_0, y_0) :

$$y_1 = y_0 + f(x_0, y_0) \Delta x \quad (18)$$

$$y_2 = y_1 + f(x_1, y_1) \Delta x \quad (19)$$

⋮

Nach dieser Herleitung könnte bei jedem Schritt ein anderer Wert für Δx gewählt werden. Das ist ohne weitere Kriterien jedoch schwierig (s.u.), daher wählt man hier für alle Schritte denselben Δx -Wert.

Bei der Euler-Integration (Trapezregel) haben wir eine lineare Interpolation zwischen zwei bekannten Punkten des Integranden gemacht. Hier machen wir eine lineare Extrapolation von einem bekannten Punkt zu einem unbekanntem, und behaupten dabei, daß die Steigung $f()$ im jeweiligen Intervall konstant ist. Dadurch brechen wir die Taylorreihe

$$f(x_{i+1}) = f(x_i) + hf'(x_i) + \frac{h^2}{2}f''(x_i) + \dots \quad (20)$$

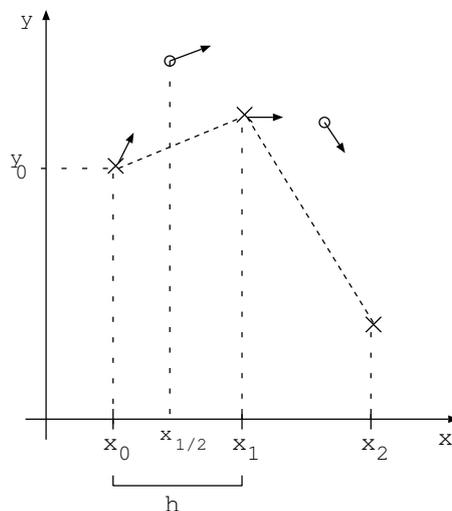
bereits nach dem Glied 0.Ordnung ab, machen also die Approximation $f = dy/dx \approx \Delta y/\Delta x$. Die Trapezregel approximiert dagegen $f' = df/dx \approx \Delta f/\Delta x$ und bricht die Taylorreihe daher erst nach dem Glied 1.Ordnung ab. Deshalb ist der Fehler hier größer:

$$\mathcal{O}(h) = \mathcal{O}(\Delta x) \quad (21)$$

Runge-Kutta-Verfahren

Man kann eine Verringerung des Fehlers erreichen, wenn man für die Steigung im Intervall nicht die Steigung am Anfang, sondern die Steigung in der Intervallmitte ansetzt, nach folgender Strategie:

- konstruiere einen Hilfspunkt in der Intervallmitte: $x_{1/2} = x_0 + \frac{1}{2}h$
- berechne den dortigen y -Wert: $y_{1/2} = y_0 + \frac{1}{2}hf(x_0, y_0)$
(bis hierher ist das ein Euler-artiger Schritt mit $h/2$)
- die Steigung in $(x_{1/2}, y_{1/2})$ ergibt sich durch Einsetzen in $f()$: $f(x_{1/2}, y_{1/2})$
- mit dieser Steigung geht man dann von x_0 nach x_1 :
 $y_1 = y_0 + hf(x_{1/2}, y_{1/2})$
(das ist wieder ein Euler-artiger Schritt, diesmal mit der Schrittweite h)



In der in diesem Gebiet üblichen Notation führt man zwei Hilfsgrößen k_1 und k_2 ein und schreibt dies als:

$$k_1 = hf(x_i, y_i) \quad (22)$$

$$k_2 = hf\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1\right) \quad (23)$$

$$y_{i+1} = y_i + k_2 \quad (24)$$

Dieses Schema heißt Runge-Kutta-Verfahren 2.Ordnung (modifiziertes Euler-Verfahren, midpoint-Verfahren). Die Erhöhung der Fehlerordnung auf

$$\mathcal{O}(h^2) \quad (25)$$

erkaufte man sich dabei durch 2 Funktionsberechnungen pro Schritt (und zwei Euler-artige Schritte pro eigentlicher Schrittweite h).

Verfahren dieser Art gibt es auch in noch höheren Ordnungen. Besonders gebräuchlich ist die Runge-Kutta-Methode 4.Ordnung, die mit der Simpson-Regel bei der Integration verwandt ist:

$$k_1 = hf(x_i, y_i) \tag{26}$$

$$k_2 = hf(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1) \tag{27}$$

$$k_3 = hf(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2) \tag{28}$$

$$k_4 = hf(x_i + h, y_i + k_3) \tag{29}$$

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{30}$$

Der Preis für den günstigen Fehler der Ordnung $\mathcal{O}(h^4)$ ist die Notwendigkeit, pro Schritt 4 Funktionsberechnungen ausführen zu müssen.

Bei vielen Differentialgleichungen in der Praxis gilt

$$h \text{ bei Runge-Kutta 4.Ordnung} \geq 2h \text{ bei midpoint} \geq 4h \text{ bei Euler}$$

Man hat also eine gute Chance, daß der erhöhte Aufwand für 4 Funktionsberechnungen pro Schritt durch die Möglichkeit, eine viel längere Schrittweite verwenden zu können, überkompensiert wird.

Wie bei der Integration gilt auch hier:

- höhere Ordnung = höhere Genauigkeit (bei vergleichbarem Aufwand) ist nur bei hinreichend gutartigen Differentialgleichungen möglich;
- eine einzige Lösung mit einem Wert von h reicht nicht; man muß durch mehrere Wiederholungen mit jeweils kleinerem h -Wert Konvergenz erreichen;
- man kann die Schrittweite h auch *zu* klein machen: durch die begrenzte Computergenauigkeit entsteht bei jeder Rechenoperation ein Fehler, und diese Fehler akkumulieren sich bei der Propagation. Dadurch kann eine Genauigkeitserhöhung, die immer durch mehr Rechenschritte erkauft werden muß, völlig zunichte gemacht werden oder sogar ins Gegenteil umschlagen.

Eigentlich sollte man Differentialgleichungen jedoch nicht mit fester Schrittweite propagieren (jedenfalls nicht bei Verwendung einer Bibliotheksroutine). Dann bestimmt nämlich die *ungünstigste* Region die Schrittweite im Gesamtintervall.

Implementation

Algorithmen zur Lösung von Differentialgleichungen mit fester Schrittweite sollten in drei Modulen implementiert werden, sehr ähnlich zur Integration:

1. Funktion **C** zur Berechnung von $f(x, y)$:

- input: x, y
- berechne output $y' = f(x, y)$

2. subroutine **B**, die einen Schritt der Länge h macht:

- input: h, x_i, y_i
- Formel zur Berechnung von y_{i+1} aus y_i, x_i, h (und ggf. k_1, k_2, \dots) und einem (oder mehreren) Aufrufen der Funktion **C**
- output: $y_{i+1}, x_{i+1} = x_i + h$

3. Hauptprogramm **A**:

- user-interface: Eingabe von
 - Anfangsbedingungen: x_0, y_0
 - x_f , sowie entweder Schrittweite h oder Schrittzahl n (die jeweils andere Größe kann/muß intern berechnet werden)
- Stelle **X**: Schleife über n Aufrufe der subroutine **B**
- Ausgabe von $y_f = y(x_f)$ und ggf. analytischer Vergleichswert
- wenn Genauigkeit nicht ausreichend, h verkleinern (bzw. n erhöhen) und zurück zu Stelle **X**

Aufgabe zur numerischen Integration

Implementieren Sie anhand des besprochenen Schemas ein Programm zur Lösung gewöhnlicher Differentialgleichungen nach Euler und/oder nach Runge-Kutta (4.Ordnung) mit fester Schrittweite. Testen Sie das Programm mit Differentialgleichungen Ihrer Wahl.

Vorschlag für den allerersten Test:

$$\frac{dy}{dx} = x \times y \quad (31)$$

mit der analytischen Lösung

$$y = A \times e^{x^2/2} \quad (32)$$

Mit dem Anfangswert $y = 1$ an der Stelle $x = 0$ ist $A = 1$ und Sie können y analytisch mit Hilfe der Zahl $e = \exp(1)$ an einer beliebigen Stelle berechnen, z.B. bei $x = 1$ zu $\sqrt{e} = 1.648721271$ (in diesem Fall werden Sie z.B. mit dem Euler-Verfahren etwa 1000 Schritte für eine Genauigkeit von wenigen Dezimalstellen benötigen).

Zur Vereinfachung der folgenden Beispiele ist es praktisch, das Programm so zu gestalten, daß es die Integration mit verdoppelter Schrittweite solange wiederholt, bis eine vorgegebene, relative Genauigkeit $\epsilon \times y$ erreicht ist. Vorsicht! Abhängig von der Genauigkeit (single oder double precision) darf ϵ nicht zu klein gewählt werden.

Atomare Einheiten (atomic units, a.u.)

Symbol	Physikalische Größe	Wert in a.u.	Wert in SI-Einheiten
m_e	Masse des Elektrons	1	$9.110 \cdot 10^{-31}$ kg
e	Elementarladung	1	$1.602 \cdot 10^{-19}$ C
\hbar	Impuls ($\hbar = h/2\pi$)	1	$1.055 \cdot 10^{-34}$ Js
h	Plancksches Wirkungsquantum	2π	$6.626 \cdot 10^{-34}$ Js
$4\pi\epsilon_0$	Elektrische Feldkonstante	1	$1.113 \cdot 10^{-10}$ C ² /J m
c	Lichtgeschwindigkeit	137.036	$2.998 \cdot 10^8$ m/s
μ_0	Magnetische Feldkonstante $4\pi/c^2$	$6.692 \cdot 10^{-4}$	$1.257 \cdot 10^{-6}$ Ns ² /C ²

Vorteile:

- weniger Konstanten in den Gleichungen (keine Faktoren \hbar^2/m_e in der kinetischen Energie, kein $e^2/4\pi\epsilon_0$ in Coulomb-Termen)
- höhere Genauigkeit in numerischen Berechnungen (typische Zwischenresultate variieren nicht über mehrere Dutzend Größenordnungen)
- Resultate unabhängig von Genauigkeit der Maßeinheiten-Bestimmung

Achtung:

- Masse: praktischere Umrechnung:
 $1822.88734 m_e = 1 \text{ amu} = 1 \text{ u} = m(^{12}\text{C})/12$
- atomare Längeneinheit:
 $1 \text{ bohr} = 1 a_0 = 5.2917742924 \cdot 10^{-11} \text{ m} = 0.52917742924 \text{ \AA}$
- atomare Zeiteinheit:
 $1 \text{ atu} = 2.41888428 \cdot 10^{-22} \text{ fs} = 2.41888428 \cdot 10^{-17} \text{ s}$
- atomare Energieeinheit:
 $1 \text{ hartree} = 1 E_h$

Energieumrechnungstabelle:

	cm^{-1}	kJ/mol	kcal/mol	eV	E_h
1 $\text{cm}^{-1} =$	1.0	$1.196266 \cdot 10^{-2}$	$2.859144 \cdot 10^{-3}$	$1.239852 \cdot 10^{-4}$	$4.556335 \cdot 10^{-6}$
1 $\text{kJ/mol} =$	83.59347	1.0	$2.390057 \cdot 10^{-1}$	$1.036436 \cdot 10^{-2}$	$3.808798 \cdot 10^{-4}$
1 $\text{kcal/mol} =$	349.7551	4.184	1.0	$4.336445 \cdot 10^{-2}$	$1.593601 \cdot 10^{-3}$
1 $\text{eV} =$	8065.479	96.48455	23.06036	1.0	$3.674902 \cdot 10^{-2}$
1 $E_h =$	219474.7	2625.5	627.5095	27.211608	1.0
1 $\text{K} =$	0.695009	0.00831433	0.00198717	$8.61707 \cdot 10^{-5}$	$3.16669 \cdot 10^{-6}$

- für Chemiker: $1 \text{ mE}_h \approx 3 \text{ kJ/mol} \approx 0.5 \text{ kcal/mol}$
- für Physiker: $1 E_h \approx 25 \text{ eV}$
- Chemiker \leftrightarrow Physiker: $1 \text{ eV} \approx 100 \text{ kJ/mol} \approx 25 \text{ kcal/mol}$

3 Klassische Mechanik

Uns interessiert die zeitliche Bewegung eines Teilchens der Masse m in einer Raumdimension x , d.h. wir suchen die Funktion $x(t)$ (die Trajektorie). Zur numerischen Behandlung diskretisieren wir die Zeitachse in gleiche Intervalle der Länge Δt und versuchen, die unbekannte Funktion $x(t)$ an den Punkten $x_n = x(t_0 + n \times \Delta t)$ zu bestimmen, wenn einer dieser Punkte x_i gegeben ist. In der Umgebung dieses Punkts können wir $x(t)$ in eine Taylorreihe entwickeln:

$$x_{i+1} = x_i + \Delta t x'_i + \frac{(\Delta t)^2}{2} x''_i + \dots \quad (33)$$

Mit einem Fehler 2. Ordnung in Δt können wir daher die 1. Ableitung, also die Geschwindigkeit $v(t) = dx/dt$, allein durch Funktionswerte ausdrücken:

$$v_i = x'_i = \frac{x_{i+1} - x_i}{\Delta t} + \mathcal{O}((\Delta t)^2) \quad (34)$$

Um die 2. Ableitung, also die Beschleunigung $a(t) = dv/dt = d^2x/dt^2$, ebenso nur durch Funktionswerte ausdrücken zu können, betrachten wir zusätzlich die Taylorreihe für einen Schritt „rückwärts“:

$$x_{i-1} = x_i - \Delta t x'_i + \frac{(\Delta t)^2}{2} x''_i - + \dots \quad (35)$$

Nach Addition der Gln. 33 und 35 ergibt sich mit einem Fehler 3. Ordnung in Δt :

$$a_i = x''_i = \frac{x_{i+1} - 2x_i + x_{i-1}}{(\Delta t)^2} + \mathcal{O}((\Delta t)^3) \quad (36)$$

Eine einfache Umstellung von Gl. 36 liefert eine Bestimmungsgleichung für x_{i+1} , wenn x_i und x_{i-1} bekannt sind:

$$x_{i+1} = 2x_i - x_{i-1} + (\Delta t)^2 a_i \quad (37)$$

Nach Newton gilt $F = ma$, also erhalten wir:

$$x_{i+1} = 2x_i - x_{i-1} + \frac{(\Delta t)^2}{m} F_i \quad (38)$$

Die Kraft F_i am Punkt x_i ergibt sich aus dem Gradienten des Potentials („Kraftfelds“) zu:

$$F_i = - \left. \frac{dV}{dx} \right|_{x=x_i} \quad (39)$$

Gl. 37 bzw. 38 ist der sog. *einfache Verlet-Algorithmus*, einer der Standard-Algorithmen in der klassischen Moleküldynamik (MD).

Für das Beispiel zweiatomiger Moleküle AB hier im Praktikum verwenden wir zur Vereinfachung des Programms die Propagation eines einzigen Punkts der reduzierten Masse μ_{AB} mit

$$\frac{1}{\mu_{AB}} = \frac{1}{m_A} + \frac{1}{m_B} \quad (40)$$

In realen Anwendungen auf größere Moleküle propagiert man in der Regel direkt in den 3N kartesischen Koordinaten der N Atome. Dadurch werden in den Gln. 37 bzw. 38 die skalaren Größen x und a bzw. F zu Vektoren mit 3N Komponenten; alles andere bleibt unverändert.

Aufgabe klassische Mechanik:

Implementieren Sie den einfachen Verlet-Algorithmus, für einen Harmonischen Oszillator bzw. einen Morseoszillator. Einige relevante Gleichungen:
Potentiale (Kraft = 1. Ableitung des Potentials nach x)

$$V_{HO} = \frac{1}{2}kx^2 \quad , \quad V_{Morse} = D_e\{1 - \exp[-\beta(x - x_e)]\}^2 \quad (41)$$

Zusammenhang zwischen Kraftkonstante k des Harmonischen Oszillators (allgemeiner: 2. Ableitung im Minimum) und den Morseparametern:

$$k = 2D_e\beta^2 \quad (42)$$

Schwingungsperiode t_p des Harmonischen Oszillators:

$$\frac{1}{t_p} = \nu_0 = \frac{1}{2\pi} \sqrt{\frac{k}{m}} \quad (43)$$

Beispiel für realistische Morseparameter (OH-Radikal):

$$\begin{aligned} D_e &= 479.9 \text{ kJ/mol} \\ \beta &= 2.22 \text{ \AA}^{-1} \\ x_e &= 0.96966 \text{ \AA} \\ \text{reduzierte Masse } \mu_{OH} &= 0.94808710 \text{ amu} \end{aligned}$$

Zielstruktur des Programms:

- Zunächst alle **real**-Variablen in doppelter Genauigkeit verwenden.
- Einlesen der nötigen input-Parameter aus einem file:
 - Potentialparameter **de,beta,xe** in kJ/mol, \AA^{-1} und \AA (der Parameter **k** für den Harmonischen Oszillator wird daraus nach Gl. 42 berechnet)
 - reduzierte Masse des OH-Systems in amu (*Vorsicht: \neq atomic units!*)
 - diskreter Zeitschritt Δt in fs (für erste Versuche: $\Delta t = 0.1$ fs)
 - Anzahl zu propagierender Zeitschritte (alternativ: End-Zeitpunkt)
 - Startpunkt $x(t=0)$ der Trajektorienpropagation
 - Beginn und Breite des graphischen Darstellungsintervalls für das Potential und die Kraft, in \AA (z.B.: 0.5 \AA , 2.5 \AA)
(dieses Intervall hat *keinen* Einfluß auf die Trajektorie!)
 - Auswahl des Potentials: harmonischer Oszillator oder Morseoszillator
 - optionaler Stop nach Potential- und Kraftausgabe, zu Testzwecken
- Konversion aller input-Parameter in atomare Einheiten
- Berechnung und Ausgabe (in fs) der analytischen Schwingungsperiode des harmonischen Oszillators nach Gl. 43
- (ausschließlich zur Kontrolle:) Berechnung und Ausgabe des Potentials (in kJ/mol) und der 1. Ableitung (Kraft) im interessierenden Koordinatenbereich
- Setzen Sie zum Start der Trajektorie $x_1 = x_0$ (Was bedeutet das physikalisch?!)
- Schleife über alle Zeitschritte; darin Verlet-Algorithmus (eine einzige Programmzeile!), Ausgabe der jeweils aktuellen Werte für Zeit (in fs) und Ort (in \AA) in ein file.

Experimente mit klassischer Mechanik:

1. **Verständnis des outputs:** Lassen Sie das Programm laufen (für den harmonischen Oszillator und für den Morseoszillator) und plotten Sie das Potential und die Kraft zusammen in einer Darstellung (ggf. den Darstellungsbereich auf der y-Achse begrenzen!), sowie die Trajektorie separat.
 - a) Verifizieren Sie qualitativ, daß die Kraft die 1. Ableitung des Potentials nach dem Ort ist.
 - b) Betrachten Sie die Trajektorie (Ort x eines fiktiven Punktes mit einer konstruierten („reduzierten“) Masse als Funktion der Zeit t) und vergegenwärtigen Sie sich, wie sich das OH-Molekül (eigentlich zwei Massenpunkte!) tatsächlich bewegt. (Rotiert es? Begründung!)
 - c) Worin unterscheidet sich die Trajektorie im Morseoszillator qualitativ von der im harmonischen Oszillator?
 - d) Berechnen Sie zusätzlich zur Trajektorie auch noch an jedem Zeitpunkt die Gesamtenergie $E = 1/2 \times \mu v^2 + V(x)$, wobei Sie die Geschwindigkeit v aus Gl. 34 ermitteln. Plotten Sie die Gesamtenergie als Funktion der Zeit. Eigentlich sollte die Gesamtenergie konstant bleiben. Verkleinern Sie den Zeitschritt, bis dies hinreichend gut zutrifft. (*Achtung:* Beim einfachen Verlet-Algorithmus ist die Geschwindigkeit nur ungenauer bekannt als die Trajektorie selber, s.o.. Verifizieren Sie dies weiter unten bei der Betrachtung der numerischen Schwingungsperiode im Vergleich zur analytischen.)

2. Schwingungsperiode:

- a) An welchen Stellen der Trajektorie kehrt sich die Bewegungsrichtung um? (Umkehrpunkte) Am wievielten Umkehrpunkt ist die erste Schwingungsperiode beendet? Warum beginnt die Trajektorie genau in einem Umkehrpunkt? Wie könnte man erreichen, daß sie an einer beliebigen anderen Stelle anfängt? Wie müßte man dazu das Programm ändern?
- b) Wie können Sie die Schwingungsperiode aus der Trajektorie ermitteln? Stimmt dieser (ungefähre) numerische Wert beim harmonischen Oszillator mit dem ausgegebenen analytischen Wert überein?
- c) Verifizieren Sie, daß die (numerische) Schwingungsperiode im harmonischen Oszillator nicht(!) vom Startpunkt der Trajektorie abhängt, obwohl der Startpunkt einen erheblichen Einfluß auf die Gesamtenergie, die Schwingungsamplitude, die Maximalgeschwindigkeit, usw., hat. Wie läßt sich das (qualitativ) verstehen?
- c) Die (hier nur numerisch verfügbare) Schwingungsperiode im Morseoszillator ist länger als im harmonischen Oszillator. Warum?
- d) Überprüfen Sie, ob sich die (numerische) Schwingungsperiode im Morseoszillator mit der Wahl des Startpunkts ändert. Wie läßt sich das Resultat qualitativ verstehen?

3. Einfluß der Größe des Zeitschritts:

- a) Verwenden Sie eine der bisherigen Trajektorien als Referenz. Vergrößern Sie dann den Zeitschritt Δt . Praktischerweise sollten Sie die Schrittzahl so mit verändern, daß Sie auf dieselbe Gesamtpropagationszeit kommen, damit Sie die veränderte und die Referenztrajektorie direkt übereinander plotten können. Bei welchen

Werten des Zeitschritts beobachten Sie welche Veränderungen? Irgendwann „explodiert“ die Propagation, d.h. Sie erhalten extrem große, offensichtlich unsinnige Werte. Warum?

- b) Verändern Sie das Programm so, daß alle reellen Variablen nur noch mit 4 statt 8 Bytes dargestellt werden. Verkleinern Sie nun sukzessive den Zeitschritt. Welche Veränderung in der Trajektorie beobachten Sie bei welchem Zeitschrittwert? Warum?

(Denselben Effekt hätten wir auch bei 8 Bytes beobachten können, allerdings erst bei so kleinen Zeitschritten, daß die Propagationen hin zu signifikanten Ortsveränderungen sehr lange dauern würden.)

- c) Beurteilen Sie aus den beiden vorigen Tests, ob der anfänglich vorgegebene Zeitschritt wirklich gut gewählt war, unter den Aspekten Genauigkeit und Propagationseffizienz.

- d) Der ideale Zeitschritt ist keine Universalie des Algorithmus, sondern systemabhängig. „Raten“ Sie sinnvoll aus Gl. 38 oder 43, wie er sich bei anderen Werten für die Masse oder die Kraft (Potentialgefälle) ändern müßte. (Wenn Sie Zeit haben, können Sie das auch empirisch überprüfen!)

Daraus ergibt sich, daß man bei MD-Anwendungen in der Chemie eigentlich immer den Zeitschritt justieren müßte – allerdings sind die Systemunterschiede selten sehr groß, daher schwanken die idealen Zeitschrittwerte nur wenig; 0.1–1.0 fs sind durchaus typische Werte.

4. Einfluß der Anzahl der Propagationsschritte:

Benutzen Sie die ungenauere 4-Byte-Version des Programms für ein weiteres wichtiges Experiment:

- a) Erzeugen Sie (mit einem guten Wert für den Zeitschritt, z.B. 0.1 fs) eine längere Trajektorie für den Harmonischen Oszillator. Lassen Sie sich sowohl die numerischen Umkehrpunkte und -zeiten als auch die entsprechenden Vielfachen der analytischen Schwingungsperiode ausgeben. Eigentlich sollten alle Schwingungsperioden exakt gleich lang sein (warum?). Tatsächlich sieht man mit der Zeit anwachsende Abweichungen (beachten Sie, daß die Schwingungsperiode kein Vielfaches des Zeitschritts ist, deshalb trifft die Trajektorie auch am Anfang die Umkehrpunkte nie ganz genau!, also können wir hier nur auf grobe Effekte schauen). Nach wievielen Zeitschritten ist der Fehler im Umkehrpunkt auf etwa 10 Zeitschritte angewachsen? (Beachten Sie, daß sich gleichzeitig auch im maximal erreichten x-Wert Fehler einzuschleichen beginnen.)

(Bei der 8-Byte-Version ist das Verhalten qualitativ genauso, nur braucht man wiederum längere Trajektorien, um dieselben Effekte zu sehen.)

- b) Woher kommt dieser Fehler? Läßt er sich durch einen kürzeren Zeitschritt vermeiden? Untersuchen Sie denselben Effekt mit einem um einen Faktor 10 kürzeren Zeitschritt (also z.B. 0.01 fs statt vorher 0.1 fs). Warum hilft ein kürzerer Zeitschritt hier tatsächlich nichts, wenn man dieselbe Endzeit erreichen will, obwohl doch der Fehler pro Schritt bei kleineren Schritten tatsächlich kleiner wird?

Durch diesen Effekt ist die maximale Anzahl von Zeitschritten und damit die maximale Propagationsdauer in MD-Simulationen sehr viel stärker beschränkt als es chemisch wünschenswert wäre: Typische Molekülbewegungen erfordern Zeitschritte im Bereich 0.1–1.0 fs; bei größeren Systemen (Biochemie) können sich interessante Vorgänge jedoch durchaus im Bereich von Millisekunden oder länger abspielen. Nach

den dafür mindestens nötigen 10^{12} Zeitschritten hat sich die simulierte Trajektorie zwangsläufig weit von der realen entfernt (auch wenn die Dynamik gar nicht chaotisch wäre!, womit realiter kaum zu rechnen ist; (deterministisches) Chaos verstärkt diesen Effekt noch einmal erheblich).

Wenn man keine reale Trajektorie reproduzieren will, sondern nur nach der statistischen Thermodynamik einen Ensemblemittelwert durch einen Zeitmittelwert approximieren will, kann man sich dennoch derart „zu lange“ Propagationszeiten erlauben, da dann die Divergenz der simulierten Trajektorie von der realen weniger wichtig ist als die (immer noch gegebene) vergleichsweise kurzzeitig korrekte Erfüllung der klassisch-mechanischen Bewegungsgesetze.

4 Lösung der zeitunabhängigen Schrödingergleichung durch Basisentwicklung

Die Schrödingergleichung in Operator/Funktionen-Form

$$\hat{H}\psi = E\psi \quad (44)$$

kann durch Entwicklung von ψ in eine Funktionen-Basis $\{\phi_i\}$ endlicher Größe (finite basis representation FBR, Spektraldarstellung)

$$\psi = \sum_{i=1}^n c_i \phi_i \quad (45)$$

in eine Matrix-Vektor-Eigenwertgleichung überführt werden:

Einsetzen der Entwicklung Gl. 45 in Gl. 44 liefert:

$$\sum_i c_i \hat{H} \phi_i = E \sum_i c_i \phi_i \quad (46)$$

Multiplikation von links mit $\langle \phi_j |$ ergibt:

$$\sum_i c_i \langle \phi_j | \hat{H} | \phi_i \rangle = E \sum_i c_i \langle \phi_j | \phi_i \rangle \quad (47)$$

Wenn die Basis $\{\phi_i\}$ orthonormal ist, gilt für die Überlappungsmatrix

$$S_{ji} = \langle \phi_j | \phi_i \rangle = \delta_{ij} \quad (48)$$

(beachte: offensichtlich sind dadurch alle Eigenwerte von S gleich Eins!)

und die Summe auf der rechten Seite kollabiert daher zu einem einzigen Term:

$$\sum_i c_i \langle \phi_j | \hat{H} | \phi_i \rangle = E c_j \quad (49)$$

Für jeden Wert von j gibt es eine solche Gleichung; diese können zusammengefaßt werden zu der Matrix-Vektor-Gleichung

$$\mathbf{H}\vec{c} = E\vec{c} \quad (50)$$

Für eine FBR-Basis aus n Funktionen erhält man zwangsläufig eine $n \times n$ -Matrix \mathbf{H} und damit genau n Eigenwerte (die nicht alle verschieden sein müssen) und dazugehörige Eigenvektoren. Dementsprechend gibt es n verschiedene Matrix-Vektor-Gleichungen Gl. 50, die man abkürzend zu einer einzigen Matrix-Matrix-Gleichung zusammenfassen kann:

$$\mathbf{H}\mathbf{C} = \mathbf{C}\mathbf{E} \quad (51)$$

Dabei sind die Vektoren \vec{c} die Spaltenvektoren der Matrix \mathbf{C} , und die Diagonalmatrix \mathbf{E} enthält die n Eigenwerte E auf der Diagonalen.

Beachte: Gl. 50 erhält man auch, wenn man $\langle \psi | \hat{H} | \psi \rangle$ durch Variation der Entwicklungskoeffizienten c_i minimiert, mit der Nebenbedingung $S_{ij} = \delta_{ij}$ (lineares Variationsproblem); diese beiden Sichtweisen sind äquivalent (Szabo/Ostlund, Kap. 1.3).

⇒ Die erhaltenen Eigenwerte sind obere Schranken zu den exakten Eigenwerten.

Beispiel für orthogonale Basis: Sinus-Funktionen

Die Eigenfunktionen eines Teilchens in einem eindimensionalen Kasten mit unendlich hohen Potentialsprüngen an den Rändern des Intervalls $[a, b]$ sind Sinus-Funktionen, die Vielfache einer halben Wellenlänge im Intervall $[a, b]$ haben und an den Intervallrändern (sowie außerhalb) Null sind:

$$\phi_m(x) = \sqrt{\frac{2}{b-a}} \sin \left\{ \frac{m\pi}{b-a}(x-a) \right\} \quad \text{für } x \in [a, b] \quad (52)$$

$$\phi_m(x) = 0 \quad \text{für } x < a \quad \text{und} \quad x > b \quad (53)$$

Entwicklung einer gegebenen reellen Funktion in diese Basisfunktionen entspricht einer (reellen) Fourieranalyse (mit der Einschränkung, daß die zu entwickelnde Funktion an den Intervallenden auch gegen Null gehen muß, ebenso wie die Basisfunktionen). In diesem Sinne sind diese Basisfunktionen sehr allgemein, ähnlich allgemein wie die nur wenigen Einschränkungen unterworfenen Fourieranalyse.

(Die mit diesen Sinus-Funktionen verwandten „ebenen Wellen“ $\exp(ikx)$ sind Standard-Basisfunktionen in der theoretischen Festkörperphysik und -chemie.)

Matrizelemente der kinetischen Energie können analytisch berechnet werden.

Matrizelemente der potentiellen Energie erfordern numerische Integration (oder Entwicklung des Potentials in geeignete Potential-Basisfunktionen).

Allgemeine Qualitätskriterien für Basisfunktionen:

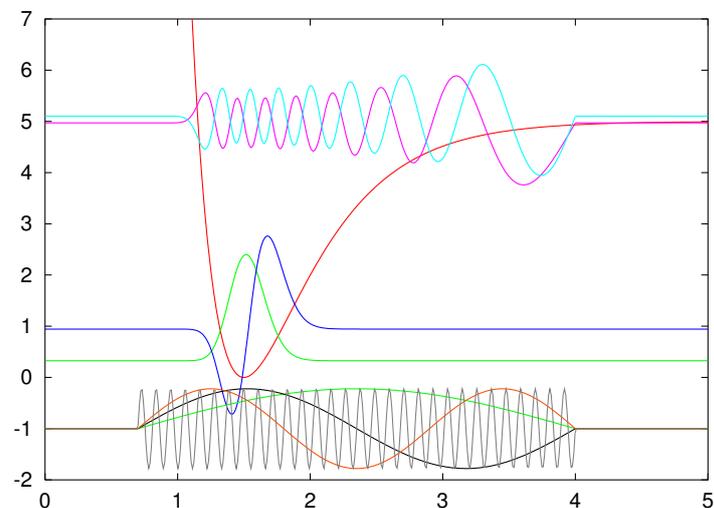
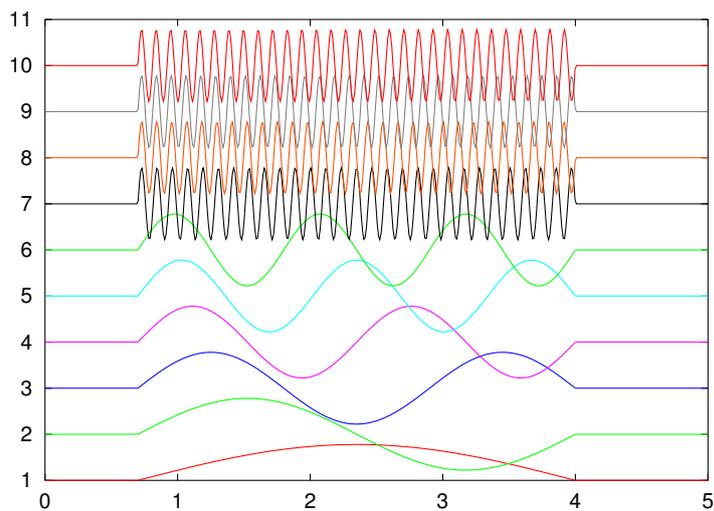
- Anzahl der Basisfunktionen muß groß genug sein;
- sie müssen mindestens den gesamten Ortsraumbereich der gewünschten Eigenfunktionen abdecken;
- sie müssen mindestens den gesamten Impulsraumbereich (Frequenzbereich der gewünschten Eigenfunktionen) abdecken;
- sie müssen passende Randbedingungen haben.

Beispiel: Sinus-Basis für 1D-Morse-Oszillator

$$\hat{H} = \hat{p}^2/2\mu + V(x), \text{ mit } V(x) = D_e\{1 - \exp[-\beta(x - x_e)]\}^2;$$

Parameter in atomaren Einheiten: $D_e = 5.0$, $\beta = 2.0$, $x_e = 1.5$, $\mu = 0.05$ amu ≈ 91.144

n	numerisch	analytisch
1	0.3257483257557883	0.3257483303645043
2	0.9443301620270594	0.9443301758532030
3	1.519025577977924	1.519025601021488
4	2.049834573608373	2.049834605869361
5	2.536757148918450	2.536757190396820
6	2.979793303908249	2.979793354603866
7	3.378943038577837	3.378943098490498
8	3.734206352927147	3.734206422056718
9	4.045583246955985	4.045583325302524
10	4.313073720748337	4.313073808227917
11	4.536677851722307	4.536677870832897
12	4.716420262499438	4.716395513117464
13	4.854119814635596	4.852226735081617
14	4.970424726778137	4.944171536725356
15	5.102696896182915	4.992229918048683



Aufgabe zeitunabhängige Quantenmechanik:

Implementieren Sie das vorgestellte Verfahren zur Lösung der zeitunabhängigen 1D-Schrödingergleichung

$$\hat{H}\psi = E\psi \quad , \quad \text{mit} \quad \hat{H} = -\frac{1}{2m}\frac{\partial^2}{\partial x^2} + V(x) \quad (54)$$

durch Entwicklung in eine Sinus-Basis, für den Morseoszillator

$$V_{Morse}(x) = D_e \{1 - \exp[-\beta(x - x_e)]\}^2 \quad (55)$$

mit Morse-Parametern für das OH-Radikal als Beispiel:

$$\begin{aligned} D_e &= 479.9 \text{ kJ/mol} \\ \beta &= 2.22 \text{ \AA}^{-1} \\ x_e &= 0.96966 \text{ \AA} \\ m_{OH} &= 0.94808710 \text{ amu} \end{aligned}$$

Zielstruktur des Programms:

- Verwenden Sie durchgängig doppelte Genauigkeit.
- Einlesen der nötigen input-Parameter aus einem file:
 - Potentialparameter: `de,beta,xe` in kJ/mol, \AA^{-1} und \AA
 - Beginn und Ende des Bereichs der numerischen Integration: `r1,r9` (gleichzeitig Anfang und Ende des Definitionsbereichs der Basisfunktionen = Intervall $[a, b]$ in Gl. 52)
 - reduzierte Masse des OH-Systems in atomic mass units (amu)
 - Anzahl Basisfunktionen: `nmax` (die Basisfunktionen sollten so konstruiert werden, daß sie $0,1,\dots,nmax-1$ Knoten haben und jede am Intervallanfang und -ende Null ist)
 - Anzahl Stützstellen in der numerischen Integration: `nint`
 - Maximalwerte für das Potential, fürs Rechnen und fürs Zeichnen, in kJ/mol (können auch unterschiedlich sein; der zweite ist rein ästhetisch; den ersten braucht man, um numerische Probleme zu vermeiden, wenn das Potential extrem groß wird, wie das Morsepotential bei $x \rightarrow 0$)
 - willkürliche Skalierungsfaktoren für die graphische Ausgabe der Basisfunktionen bzw. der Eigenfunktionen
- Einheitenumrechnung der Parameter; Berechnung von Konstanten
- Berechnung des Morsepotentials an allen `nr` Integrationspunkten; *Test*: Plotausgabe!
- Aufstellung der Matrix `g(nint,nmax)` der `nmax` Sinus-Basisfunktionen auf den `nint` Gitterpunkten. *Test*: Plotausgabe!
- analytische Berechnung der Matrix `gdg(nmax,nmax)` der kinetischen Energie (diese Matrix ist diagonal! – wieso? Was steht auf der Diagonalen?)

- numerische Berechnung der Matrix $\text{gvg}(\text{nmax}, \text{nmax})$ der potentiellen Energie durch triviale Diskretisierung des Integrals, und unter Verwendung der bereits berechneten Sinus-Basisfunktionen $\text{g}(\text{nr}, \text{nmax})$:

$$\text{gvg}(\text{j}, \text{k}) = \Delta x \sum_{i=1}^{\text{nr}} \text{g}(\text{i}, \text{j}) \text{V}(\text{i}) \text{g}(\text{i}, \text{k}) \approx \int_{-\infty}^{+\infty} g_j(x) V(x) g_k(x) dx \quad (56)$$

Test: Dabei zunächst das Potential weglassen. Dadurch berechnen Sie hier die Überlappungsmatrix der Basisfunktionen, die (mit numerischer Genauigkeit) die Einheitsmatrix sein sollte. Dadurch testen Sie (1) die Genauigkeit Ihrer numerischen Integration (ggf. Δx verkleinern!), und (2) noch einmal die korrekte Implementation der Basisfunktionen.

Danach nicht vergessen, das Potential wieder hinzuzufügen...

- Zusammensetzen der Hamiltonmatrix **ham**: Summe aus den Matrizen der kinetischen und potentiellen Energie (inkl. richtigem Vorfaktor vor der kinetischen Energie, wenn noch nicht geschehen)
- Hamiltonmatrix wird diagonalisiert durch Aufruf von **jacobi/eigsrt** oder **dsyev** (Achtung: Lösung „in place“: danach stehen die Eigenvektoren in **ham!**)
- Ausgabe aller numerischen Eigenwerte, im Vergleich zu den analytischen.
Test: Stimmen wenigstens die ersten 2–3 numerischen Eigenwerte mit den analytischen überein?
- Berechnung der Anzahl der gebundenen Zustände nach

$$n_{\text{bound}} = \text{int} \left(\frac{\sqrt{2mD_e}}{\beta} + \frac{1}{2} \right) \quad (57)$$

(wobei „int(x)“ die größte ganze Zahl kleiner als x ist)

und Ausgabe nur der gebundenen Eigenwerte (numerisch und analytisch).

Test: Vergleich aller gebundenen Eigenwerte.

- Berechnung der Eigenfunktionen durch Bildung von Linearkombinationen der Sinus-Basisfunktionen $\text{g}(\text{nr}, \text{nmax})$, mit den (richtigen!) Elementen der Eigenvektoren als Koeffizienten. (*Beachte:* Das Gesamt-Vorzeichen der Eigenfunktionen ist unbestimmt!)
Test: Plotausgabe!

Zusätzlich gibt es noch zwei kleine Unterprogramme:

vmorse: Berechnung des Potentials $V_{\text{Morse}}(x)$ nach Gl. 55 für die gegebenen Parameter, für einen beliebigen, gegebenen **x**-Wert

emorse: Auswertung der Formel zur analytischen Berechnung gebundener Eigenenergien des Morsepotentials, für eine gegebene Schwingungsquantenzahl **n**:

$$E_n = \omega \left\{ \left(n + \frac{1}{2} \right) - \frac{\omega}{4D_e} \left(n + \frac{1}{2} \right)^2 \right\} \quad (58)$$

$$\text{mit } \omega = \beta \sqrt{\frac{2D_e}{m}} \quad (59)$$

Test: Vergleichen Sie die analytischen Eigenwerte mit der graphischen Potentialausgabe und der analytischen Anzahl gebundener Eigenwerte (s.o.)!

Experimente mit dem Morseoszillator in Sinusbasis:

Nachdem Ihr Programm nach allen bisherigen Tests korrekt funktioniert, lassen Sie Ihr Programm mit einem Referenzinput laufen (u.a. mit ca. 60 Basisfunktionen, Basisfunktionsintervall 0 bis 5 Å, 500 Integrationspunkte, rechnerischer cutoff des Potentials bei 600 kJ/mol), betrachten Sie die Eigenenergien und plotten Sie das Potential zusammen mit verschiedenen Basisfunktionen und/oder Eigenfunktionen. (*Vorsicht*: Die Quantenzahl n in Gl. 58 läuft nicht ab 1 sondern ab 0 !). Bearbeiten Sie nun folgende Fragestellungen:

1. **Verständnis des outputs:** Betrachten Sie unter Punkt 1 zunächst *nur* die analytischen Eigenenergien.

- Die analytischen Eigenwerte werden ab $n=44$ negativ. Warum ist das nicht sinnvoll?
- Ab $n=23$ werden die vorher monoton ansteigenden analytischen Eigenwerte wieder kleiner – das ist nicht sinnvoll für eine Formel, die mit wachsendem n steigende Energiewerte liefern soll. Also sind die analytischen Eigenwerte sogar bereits ab $n=23$ falsch.
- Was aber ist mit $n=22$? Die analytische Energie dafür ist kleiner als D_e , also sollte ein gebundener Zustand vorliegen. Das Programm berechnet nach Gl. 57 jedoch nur 21 gebundene Zustände. Wer hat Recht?

Anleitung: Fassen Sie die analytische Formel für die Eigenenergien in Gl. 58 so auf, als wäre n eine kontinuierliche, reelle Variable. Dann ist Gl. 58 eine nach unten geöffnete Parabel. Bestimmen Sie analytisch und graphisch, bei welchem (reellen) Wert von n sie ihr Maximum hat. Vergleichen Sie das Resultat mit Gl. 57 und dem Programm-output. Liegt das Maximum vor oder nach 22? Was folgt daraus für den Status der analytischen Eigenenergie für $n=22$?

(Bei Verständnisschwierigkeiten empfiehlt sich ein Plot der analytischen Energie als Funktion des kontinuierlichen, reellen n , mit Kennzeichnung der Energien bei ganzzahligen n -Werten. Wiederum Vorsicht mit evtl. unterschiedlichen Numerierungsweisen! (ab 0 bzw. ab 1))

2. **Gebundene und ungebundene Zustände:**

- Wieso gibt es mehr numerische Eigenwerte als gebundene Zustände? Wonach richtet sich die Anzahl numerischer Eigenwerte?
- Wie müßte eine ungebundene Wellenfunktion (mit einer Eigenenergie oberhalb der Dissoziationsschwelle D_e) aussehen? Vergleichen Sie Ihre Vorstellung mit dem, was das Programm produziert. Haben die numerischen Eigenwerte/-funktionen oberhalb der Dissoziationsschwelle D_e (ab $n=22$) eine Bedeutung? Wie müßten Ihre Basisfunktionen aussehen, damit Sie ungebundene Zustände richtig berechnen können? Welches Problem bringt das mit sich? (Hinweis: Integrationsgrenzen in Gl. 56!)
- Plotten Sie die (numerischen!) Energieeigenwerte als Funktion von n . Betrachten Sie die Energiedifferenz aufeinanderfolgender Zustände; nimmt diese mit steigender Energie ab oder zu? Offenbar gibt es zwei verschiedene Bereiche: Unterhalb und oberhalb von (ungefähr) D_e . Oberhalb von D_e ähnelt das Eigenenergiespektrum dem eines Teilchens im Kasten. Inwiefern und wieso? (Hilfe: Wie sehen die Eigenfunktionen in diesem Bereich nahe der Obergrenze b ihres Definitionsbereichs, genau an der Grenze und jenseits davon aus?)

3. **Minimale Basis:** Wenn die Berechnung von ungebundenen Eigenzuständen hier eigentlich sinnlos ist, kann man sich gezielt nur auf die Berechnung gebundener Zustände beschränken? Probieren Sie als Anzahl von Basisfunktionen etwa $n_{\max}=21$ oder noch kleinere Werte.

- a) Vergleichen Sie die relativen Fehler der numerischen Eigenwerte Ihrer bisherigen Rechnungen (mit ca. 60 Basisfunktionen) mit den jetzigen. Die Fehler sollten jetzt für alle Eigenwerte deutlich größer sein.
- b) Überlegen Sie sich, warum das so sein könnte, *bevor* Sie sich die Eigenfunktionen ansehen.
- c) Plotten Sie jetzt die Eigenfunktionen, zunächst nur für $n=1$, zusammen mit der Potentialkurve. Innerhalb des Potentialtopfs sieht die Eigenfunktion qualitativ richtig aus. Was ist außerhalb qualitativ falsch? Woher kommt das? Können Sie diese Fehler auch bei den Eigenfunktionen entdecken, die Sie vorher mit dem ursprünglichen Parametersatz (60 Basisfunktionen) gerechnet haben?
- d) Plotten Sie die Eigenfunktion zu $n=1$ für diesen und den vorigen Parametersatz direkt übereinander. Ist die neue Eigenfunktion innerhalb des Potentialtopfs qualitativ richtig?
- e) Wie müßten die Basisfunktionen aussehen, damit man mit k Basisfunktionen die untersten k gebundenen Zustände exakt erhält? Beginnen Sie die Überlegung mit dem (eigentlich sinnlosen) Fall $k = 1$, betrachten Sie dann $k = 2$ und verallgemeinern Sie.

Was ist vor diesem Hintergrund von der „minimal basis“ zu halten, die diese Eigenschaft hatte (k Basisfunktionen für k Eigenwerte) und früher in der Quantenchemie verwendet wurde?

4. Basisfunktionsintervall im Impulsraum:

- a) Vergleichen Sie nun für $n=2,3,\dots$ die Eigenfunktionen paarweise, zwischen dem neuen und dem alten Parametersatz (60 bzw. 21 Basisfunktionen). (Ignorieren Sie unterschiedliche Gesamtvorzeichen der Eigenfunktionen – diese liegen ohnehin nicht fest!) Plotten Sie als Hilfe zusätzlich die höchste Basisfunktion dazu. Welcher weitere, qualitative Fehler in der Form der Eigenfunktionen schleicht sich ab ca. $n=3$ ein und wird bei $n=4,5$ immer offensichtlicher? (*Hinweis:* Achten Sie auf die Frequenzen der Funktionen!) Was ist der Grund dafür? (*Hinweis:* Vergleichen Sie mit der Frequenz der höchsten Basisfunktion!)

Gehen Sie wieder zurück zum ursprünglichen Parametersatz (60 Basisfunktionen) und verändern Sie jetzt das Ortsraumintervall $[a, b]$ (input-Parameter $r1$ und $r9$) von 0.0–5.0 auf 0.5–4.0.

- b) Betrachten Sie die gebundenen Eigenwerte. Gegenüber dem ursprünglichen Parametersatz sind alle gebundenen Eigenenergien besser geworden (= dichter an den analytischen Werten), sogar der unterste Zustand. Das kann nicht direkt am kleineren Ortsraumbereich liegen: z.B. ist der $n = 1$ -Zustand nur im Intervall von ca. 0.65–1.45 graphisch von Null verschieden, also sollte alles, was außerhalb passiert, für diesen Zustand egal sein. Außerdem hätte man denken können, daß einige oder sogar viele Eigenenergien schlechter werden, wenn der Ortsraumbereich, den die Basisfunktionen abdecken, kleiner wird; offensichtlich ist das aber überhaupt nicht der Fall. Was ist der Grund für die durchgängige Verbesserung? (*Hinweis:* das hat mit (4a) zu tun!)

- c) *Zusatzüberlegung für reine Theoretiker:* Die Eigenfunktion eines freien Teilchens ist im Wesentlichen $\exp(ikx) = \cos(kx) + i \sin(kx)$, mit der Wellenzahl k , die direkt mit dem Impuls zusammenhängt: $p = \hbar k$. Zeigen Sie, daß sich k aus der kinetischen Energie E_{kin} nach $k = \sqrt{2mE_{kin}}$ berechnen läßt. Wenn man „mit ausreichend hoher Vergrößerung hinschaut“, ist jedes Potential lokal nahezu konstant, also kann man den „lokalen Impuls“ einer Wellenfunktion aus E_{kin} an einem Punkt berechnen.

Berechnen Sie k_{max} für das vorliegende Morsepotential. Nach Gl. 52 muß gelten: $k = m\pi/(b - a)$. Berechnen Sie danach den für k_{max} nötigen Wert m_{max} .

Ergebnis: für das Intervall 0.0–5.0 gilt $m_{max} > 60$, für das Intervall 0.5–4.0 dagegen $m_{max} < 60$.

Was folgt daraus?

5. Basisfunktionsintervall im Ortsraum:

- a) Bei genauer Betrachtung der Eigenwerte stellt sich heraus, daß der Zustand $n=21$ numerisch ungebunden ist (Eigenenergie oberhalb von D_e), analytisch aber gebunden sein sollte. Wenn seine Eigenenergie denselben Fehler hätte wie den von $n=20$ oder 19, wäre er auch numerisch noch gebunden. Also gibt es bei $n=21$ eine zusätzliche Fehlerquelle, die bei 20,19,... nicht wesentlich ist. Welche ist das?

Hinweis: Plotten Sie nochmals die Eigenfunktion für $n=21$ und vergleichen Sie das Verhalten der Funktion auf dem linken und am rechten Intervallrand miteinander (Funktionswert, Steigung, Stetigkeit der 1. Ableitung). Eines von beiden ist qualitativ falsch – welches? Warum? (Natürlich können Sie auch mit dem Verhalten der Eigenfunktionen für $n=20,19,\dots$ vergleichen!)

- b) Vergrößern Sie das Intervall an der entsprechenden Seite, bis das unerwünschte Verhalten verschwindet. Bis wohin müssen Sie gehen?

Achtung: Sie müssen nach (4b,c) *gleichzeitig* die Anzahl der Basisfunktionen erhöhen, sonst erzeugen Sie eine Verschlechterung im Impulsraum, die Ihnen den gewünschten Effekt kaputt macht.

- c) Tatsächlich reicht eigentlich bereits die nach (4c) nötige Erhöhung der Anzahl der Basisfunktionen bei Beibehaltung des Ortsraumintervalls 0.5–5.0, um $n=21$ auch numerisch gebunden zu machen. Ein Plot der zugehörigen Eigenfunktion für diesen Fall zeigt, daß das Verhalten am Rand zwar besser, aber immer noch nicht ganz richtig ist. Die Eigenenergie stimmt nur auf 3 signifikante Stellen.

- d) Versuchen Sie, durch Intervallvergrößerung und gleichzeitige, angemessene Erhöhung der Anzahl der Basisfunktionen, die Eigenenergie für $n=21$ auf 4 signifikante Stellen richtig zu treffen.

- e) Dabei werden Sie vermutlich an einen Punkt kommen, bei dem die numerische Eigenenergie *unter* die analytische sinkt. Weil das Verfahren variationell ist, sollte das „eigentlich“ nicht möglich sein. Der Grund für dieses scheinbare Paradoxon ist der Potential-cutoff: Dadurch können die höchsten Eigenfunktionen etwas weiter nach links in die Potentialwand hinein tunneln. Somit sehen sie ein effektiv zu breites Potential (breiter als das wahre Potential), wodurch die Eigenenergien etwas absinken. Korrigieren Sie in diesem Fall den cutoff (ungefähre Verdopplung). Bis wohin müssen Sie die Eigenfunktion darstellen, um unter diesen Bedingungen eine auf 4 signifikante Stellen richtige Eigenenergie zu erzielen?

Wichtige Bemerkung: Offensichtlich könnte man hier mogeln, und z.B. ein zu schmales Intervall durch einen zu niedrigen cutoff „ausgleichen“ und so durch Kompensation zweier Fehler mit etwas Geduld die höchsten Eigenenergien sehr genau treffen. Die Hauptgefahren einer solchen „Strategie“ sind:

- mangelnde Universalität: In der Eigenenergie kompensieren sich die Fehler, können sich aber in anderen Größen verstärken. Im vorliegenden Fall können Sie graphisch nachvollziehen, daß sie sich z.B. in der Wellenfunktion selber verstärken. Berechnet man andere Größen als Energie-Erwartungswerte aus der „doppelt falschen“ Wellenfunktion, ist offenbar keineswegs mit Kompensation zu rechnen!
- mangelnde Übertragbarkeit: Würden wir dasselbe für ein anderes Potential machen, könnte die Fehlerkompensation wieder erreicht werden, aber mit ziemlicher Sicherheit bei einer anderen Kombination von cutoff- und Intervallbreite-Werten.

Solche Fehlerkompensationen sind in der Computerchemie allgegenwärtig (s. Vorlesung: zu kleine Basissätze können manchmal mangelnde Elektronenkorrelation kompensieren, usw.). Dabei kommt es häufig vor, daß gute Energien (Energiedifferenzen) und sogar gute Molekülgeometrien noch nicht garantieren, daß die Wellenfunktion und andere aus ihr berechnete Eigenschaften gut sind!

6. **Numerisches Integrationsgitter:** Der letzte verbleibende Parameter `nint` (Anzahl der Integrationspunkte) konnte bisher unverändert bleiben, weil er ausreichend groß gewählt war.
 - a) Reduzieren Sie `nint` (mit dem bisher besten sonstigen Parametersatz) solange, bis sich die Eigenwerte deutlich verschlechtern (z.B. Rückgang von 4 auf 2 richtige Stellen im höchsten Eigenwert).
 - b) Ein Plot der Eigenfunktion(en) zeigt frei Auge sichtbare „Defekte“ in den Funktionen (Knicke, schlecht ausgeprägte Oszillationen). Daß die Eigenenergien weit weniger sensitiv reagieren, liegt daran, daß sie einer Integration über die Wellenfunktion entsprechen, wobei sich Fehler leicht herausheben können.
7. **Algorithmusskalierung:** Messen Sie den Zeitverbrauch des Programms mit Standard-f90-Routinen, und zwar separat für den Integrationsteil und die Diagonalisierung. Wie skalieren diese Teile mit Vergrößerung der Anzahl der Integrationspunkte und/oder mit Vergrößerung der Anzahl der Basisfunktionen?
 (Bei diesen Tests müssen Sie natürlich überhaupt nicht auf die Güte der Ergebnisse oder den Sinn der Parametersetzungen achten!)

Orthogonalisierung der Basis

Wenn die Basis $\{\phi_i\}$ nicht orthogonal ist, erhalten wir auf demselben Weg wie oben ein verallgemeinertes Matrix-Eigenwertproblem:

$$\mathbf{H}\mathbf{C} = \mathbf{S}\mathbf{C}\mathbf{E} \quad (60)$$

Dieses kann mit dafür geeigneten Bibliotheksroutinen direkt gelöst oder zunächst in ein normales Eigenwertproblem transformiert werden, indem wir die Basis mit einer geeigneten Transformationsmatrix \mathbf{X} orthogonalisieren:

$$\mathbf{X}^\dagger \mathbf{S} \mathbf{X} = \mathbf{1} \quad (61)$$

Da \mathbf{S} hermitesch (symmetrisch) ist, ist \mathbf{X} unitär.

Es gibt verschiedene Möglichkeiten für die Wahl von \mathbf{X} , eine ist:

$$\mathbf{X} = \mathbf{S}^{-1/2} = \mathbf{U}\mathbf{s}^{-1/2}\mathbf{U}^\dagger \quad (62)$$

denn dann gilt:

$$\mathbf{X}^\dagger \mathbf{S} \mathbf{X} = \mathbf{S}^{-1/2} \mathbf{S} \mathbf{S}^{-1/2} = \mathbf{S}^{-1/2} \mathbf{S}^{1/2} = \mathbf{S}^0 = \mathbf{1} \quad (63)$$

Durch diese Transformation \mathbf{X} erhalten wir auch eine neue Koeffizientenmatrix:

$$\mathbf{C}' = \mathbf{X}^{-1} \mathbf{C} \quad \Leftrightarrow \quad \mathbf{C} = \mathbf{X} \mathbf{C}' \quad (64)$$

Einsetzen von $\mathbf{C} = \mathbf{X} \mathbf{C}'$ in $\mathbf{H}\mathbf{C} = \mathbf{S}\mathbf{C}\mathbf{E}$ liefert:

$$\mathbf{H}\mathbf{X}\mathbf{C}' = \mathbf{S}\mathbf{X}\mathbf{C}'\mathbf{E} \quad (65)$$

Multiplikation von links mit \mathbf{X}^\dagger ergibt:

$$\mathbf{X}^\dagger \mathbf{H} \mathbf{X} \mathbf{C}' = \mathbf{X}^\dagger \mathbf{S} \mathbf{X} \mathbf{C}' \mathbf{E} \quad (66)$$

Dabei definiert $\mathbf{X}^\dagger \mathbf{H} \mathbf{X} = \mathbf{H}'$ die entsprechend transformierte Hamilton-Matrix. Mit Gl. 61 ergibt sich schließlich ein Standard-Eigenwertproblem:

$$\mathbf{H}' \mathbf{C}' = \mathbf{C}' \mathbf{E} \quad (67)$$

Wenn einzelne Basisfunktionen nahezu linear abhängig voneinander werden, gibt es numerische Probleme. Symptom: sehr kleine Eigenwerte von \mathbf{S} (exakt Null bei voller linearer Abhängigkeit). \Rightarrow Vor Orthogonalisierung Eigenwerte von \mathbf{S} berechnen und ggf. linear (nahezu) abhängige Basisfunktionen eliminieren.

Beispiel für nicht-orthogonale Basis: DGB

Satz von Gauß-Funktionen mit Zentren auf Ortsraumgitter: distributed Gaussian basis (DGB) ¹

$$\phi_i(x) = \left(\frac{2A_i}{\pi}\right)^{1/4} \exp[-A_i(x - x_i)^2] \quad (68)$$

Möglichkeiten für Wahl der Zentren x_i und Breiten A_i :

- plazierte Zentren x_i nach semiklassischen Kriterien und setze $A_i \sim 1/(x_i - x_{i-1})^2$
- äquidistante Zentren x_i ; optimiere Parameter A durch Minimierung der Eigenwerte in einer separaten Testrechnung

Überlappungsmatrixelemente und Matrixelemente der kinetischen Energie analytisch berechenbar (wobei T_{ij} noch den üblichen Vorfaktor $-\hbar^2/2m$ braucht):

$$S_{ij} = \sqrt{\pi} \frac{A_{ij}}{B_{ij}} \exp(-C_{ij}) \quad , \quad T_{ij} = -2S_{ij} \frac{A_i A_j}{B_{ij}^2} (1 - 2C_{ij}) \quad (69)$$

mit den Abkürzungen

$$A_{ij} = \left(\frac{4A_i A_j}{\pi^2}\right)^{1/4} \quad , \quad B_{ij} = (A_i + A_j)^{1/2} \quad , \quad C_{ij} = \frac{A_i A_j}{B_{ij}^2} (x_i - x_j)^2 \quad (70)$$

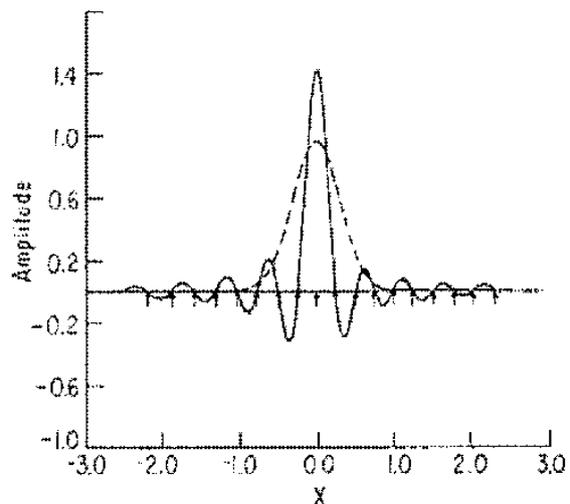
Weitere Standard-Möglichkeit zur Einsparung von Rechenzeit: Für hinreichend nicht-pathologische Operatoren \mathcal{O} gilt

$$\langle \phi_i | \mathcal{O} | \phi_j \rangle \approx 0 \quad (71)$$

wenn nur die Zentren x_i und x_j der beiden Gaußfunktionen weit genug auseinander liegen (in der Praxis schon bei 5–10 dazwischenliegenden Gaußfunktionen der Fall).

Basis-Qualitätskriterien wie bei Sinus-Basis.

Beispiel für eine DGB-Funktion vor (gestrichelte Linie) und nach (durchgezogene Linie) Orthogonalisierung



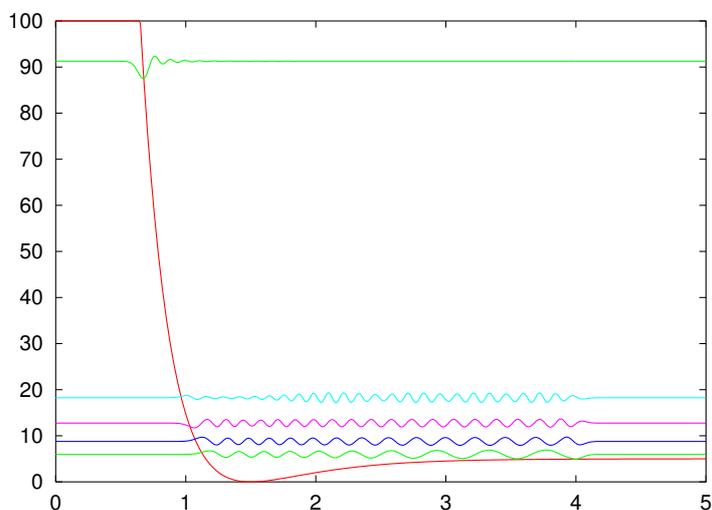
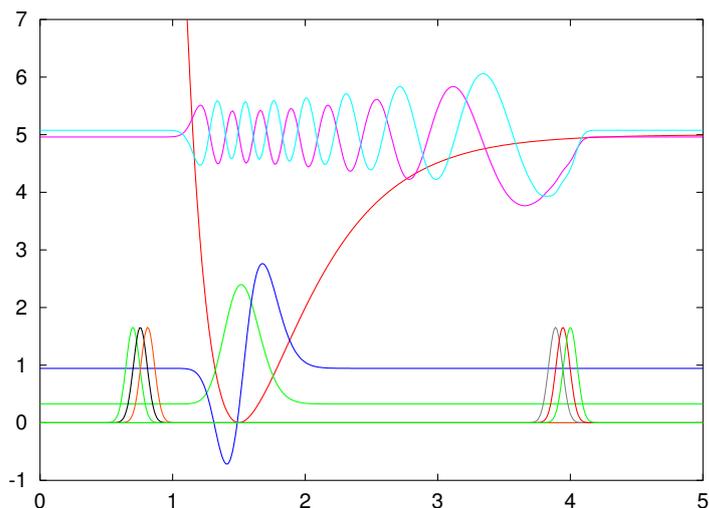
¹ I. P. Hamilton und J. C. Light, J. Chem. Phys. 84 (1986) 306; Z. Bačić und J. C. Light, J. Chem. Phys. 85 (1986) 4594; Z. Bačić, D. Watt und J. C. Light, J. Chem. Phys. 99 (1988) 947.

Beispiel: DGB für 1D-Morse-Oszillator

$$\hat{H} = \hat{p}^2/2\mu + V(x), \text{ mit } V(x) = D_e\{1 - \exp[-\beta(x - x_e)]\}^2;$$

Parameter in atomaren Einheiten: $D_e = 5.0$, $\beta = 2.0$, $x_e = 1.5$, $\mu = 0.05$ amu ≈ 91.144

n	numerisch	analytisch
1	0.3257483281296143	0.3257483303645043
2	0.9443301695380278	0.9443301758532030
3	1.519025592324165	1.519025601021488
4	2.049834599285713	2.049834605869361
5	2.536757195227862	2.536757190396820
6	2.979793386442639	2.979793354603866
7	3.378943178632529	3.378943098490498
8	3.734206573395122	3.734206422056718
9	4.045583564466534	4.045583325302524
10	4.313074135778425	4.313073808227917
11	4.536678273982599	4.536677870832897
12	4.716401680901365	4.716395513117464
13	4.852983810524478	4.852226735081617
14	4.960405968478248	4.944171536725356
15	5.076871557803291	4.992229918048683



Das Programm „morse_DGB“

Dieses Programm ist eine leicht modifizierte Variante von „morse_sinus“. Es löst ebenfalls die zeitunabhängige 1D-Schrödingergleichung für denselben HO-Morseoszillator, aber nicht mit Sinusfunktionen als Basis, sondern mit verteilten Gaußfunktionen (DGB).

Dazu bedurfte es nur weniger Modifikationen im Programm:

- Vorgabe eines äquidistanten Punktgitters als Zentren der Gaußfunktionen: neue input-Parameter „`dg0,dg9`“ = Anfang und Ende dieses Gitters (mit der gegebenen Gesamtzahl `nmax` der Gaußfunktionen läßt sich daraus ihr Abstand `deldg` berechnen).
- geeignete Skalierung der Breite der Gaußfunktionen (bestimmt vom Faktor A im Exponenten) in Abhängigkeit des Abstands Δ zweier benachbarter Gaußfunktionen: Nach Light et al. wird hier $A = 0.59/(\Delta^2)$ verwendet (dabei ist 0.59 der input-Parameter `dga`)
- analytische Berechnung der Matrixelemente der kinetischen Energie und der Basisfunktionen-Überlappung (Matrix \mathbf{S}) nach den gegebenen Formeln Gln. 69,70
- zur Kontrolle wird die Überlappungsmatrix auch noch numerisch berechnet; die Resultate können auf Wunsch mit den analytischen verglichen werden (file „`an-num_overlap.dat`“)
- wie sich zeigt (s. nächste Seite), ist der Zusatzaufwand hilfreich, die Eigenwerte der Überlappungsmatrix \mathbf{S} zu berechnen (mit einem Aufruf von `dsyev`).
- direkte Lösung des verallgemeinerten Eigenwertproblems $\mathbf{HC} = \mathbf{ESC}$ durch die LAPACK-Bibliotheksroutine „`dsygv`“. (Diese Routine führt intern das verallgemeinerte Eigenwertproblem auf ein spezielles zurück, allerdings nicht durch explizite Basisorthogonalisierung, sondern durch eine Cholesky-Faktorisierung von \mathbf{S} , löst das spezielle Eigenwertproblem und transformiert die Lösung auf die des verallgemeinerten.)

Experimente mit „morse_DGB“:

Viele Phänomene bei der verteilten Gaußbasis sind ähnlich der bei der Sinusbasis, z.B. Auswirkungen einer Beschränkung des Intervalls im Ortsraum; sie werden daher hier nicht wiederholt. Stattdessen betrachten wir einige Effekte, die bei Sinusbasen nicht oder nur eingeschränkt auftreten:

1. **Lineare Abhängigkeit der Basisfunktionen:** Machen Sie die Gauß-Basisfunktionen sukzessive breiter, durch Erniedrigung(!) des input-Werts von `dga`. Beobachten Sie dabei sowohl die Eigenwerte der Überlappungsmatrix als auch die Energieeigenwerte. Ungefähr an dem Punkt, an dem der kleinste Eigenwert von **S** „numerisch Null“ wird (das ist in doppelter Genauigkeit ein Wert von $\epsilon \approx 1 \times 10^{-16}$; definiert als kleinster Wert, bei dem noch gilt $1 - \epsilon < 1$), treten plötzlich stark negative Energieeigenwerte mit Eigenfunktionen sinnloser Form auf. Noch etwas später versagt der Löser `dsygv` des verallgemeinerten Eigenwertproblems, weil die Matrix **S** dann numerisch singular wird. (Der Löser `dsyev` des speziellen Eigenwertproblems zur Berechnung der Eigenwerte von **S** ist so robust, daß ein Versagen kaum provozierbar ist, allerdings ist den Resultaten von `dsyev` vermutlich nicht mehr zu trauen, wenn `dsygv` versagt.)

2. **Lücken im Ortsraum:** Machen Sie umgekehrt die Gauß-Basisfunktionen sukzessive schmaler, durch Erhöhung von `dga`. Beobachten Sie wiederum die **S**-Eigenwerte, die Energieeigenwerte und die Eigenfunktionen, gleichzeitig aber auch einige benachbart liegende Basisfunktionen!

Es zeigt sich, daß bereits dann, wenn man in einer graphischen Darstellung der Basisfunktionen zu sehen glaubt, daß sich benachbarte Basisfunktionen noch recht gut überlappen, die Güte der numerischen Energieeigenwerte merklich sinkt und die Eigenfunktionen anfangen „krakelig“ auszusehen. Wenn die Basisfunktionsüberlappung im Plot sichtbar klein wird, sind sowohl die Energieeigenwerte als auch die Eigenfunktionen bereits indiskutabel schlecht (woher kommen die vielen „Spitzen“ in den Eigenfunktionen?)

Denksportaufgabe: Warum gehen dabei die (alle) Eigenwerte der Überlappungsmatrix gegen 1.0 ?

Anmerkung: Eigentlich müßte man bei schmaler werdenden Basisfunktionen irgendwann die Anzahl der Integrationsgitterpunkte erhöhen. Das ließe sich anhand des Vergleichs der numerischen mit der analytischen Überlappungsmatrix testen. Beim Vorgabewert dafür ist das jedoch noch kein Problem.

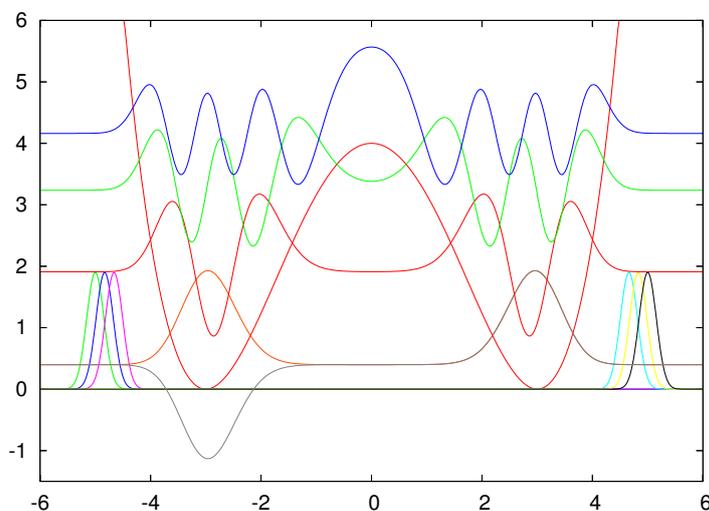
3. **Minimale Basis:** Im Gegensatz zu den Sinus-Basisfunktionen ist es hier tatsächlich möglich, mit einer relativ kleinen Basis nur einen Teil der Eigenwerte und -funktionen mit vernünftiger Genauigkeit zu berechnen: Versuchen Sie, den ersten Eigenwert auf 5 signifikante Stellen richtig zu berechnen, mit möglichst wenig Basisfunktionen in einem möglichst schmalen Intervall (erheblich kleiner als die Vorgabe).

(Ich schaffe es mit 7 Basisfunktionen — schaffen Sie es mit nur 6?!)

Beachte: Auch hier ist es bei weitem *nicht* möglich, n Eigenfunktionen mit n Basisfunktionen in sinnvoller Genauigkeit zu berechnen! Typischerweise ist die ganze obere Hälfte des Eigenwert- und Eigenfunktionenspektrums quantitativ oder sogar qualitativ falsch.

DGB-Basis im Doppelminimumpotential:

Mit minimalen Änderungen können beide Programme (mit Sinus- bzw. DGB-Basisfunktionen) auch für nahezu beliebige andere Potentiale eingesetzt werden, z.B. das DGB-Programm für ein Doppelminimumpotential:



Eine einfache analytische Form für ein symmetrisches Doppelminimumpotential ist:

$$V(x) = E_b \left[\left\{ \left(\frac{r}{r_m} \right)^4 - 2 \left(\frac{r}{r_m} \right)^2 \right\} + 1 \right] \quad (72)$$

Die obige Zeichnung wurde erzeugt mit

- Barrierenhöhe $E_b = 4.0 E_h$
- Minimumposition $r_m = 3.0 a_0$
- reduzierte Masse $\mu = 5.46866202 m_e$

(Achtung: Gezeigt sind nur die Eigenfunktionen 1,2,5,9,11.)

Obwohl Doppelminimumpotentiale in sehr vielen chemischen Situationen auftreten, handelt es sich bei dem hier gewählten Potential um ein willkürliches, „mathematisches“ Beispiel, daher haben die konkreten Zahlenwerte keine chemische Bedeutung und die Parameter werden in atomaren Einheiten eingegeben.

Wichtige Änderungen im Programm „double_wellDGB“:

Das Potential ist per Konstruktion spiegelsymmetrisch zur y-Achse, daher sind auch die Intervalle des Integrationsgitters und der Basisfunktionsplatzierung symmetrisch zum Nullpunkt auf der x-Achse. Deswegen wird der input-Parameter $r1$ (= Beginn des Integrationsgitters) nicht benötigt, da das Gitter immer von $-r9$ bis $+r9$ reicht. Analog werden Basisfunktionen immer zwischen $-dg9$ und $+dg9$ plazierte. Den input-Parameter $dg0$ gibt es trotzdem noch – er hat jedoch eine neue Bedeutung bekommen: s.u.

Experimente mit „double_well_DGB“

1. **Referenzrechnung:** Hier haben wir keine analytische Lösung, also müssen wir uns vor den ersten Tests sicher sein, daß unsere numerische Lösung korrekt ist. Verbessern Sie die vorgegebenen Parameter (Anzahl der DGB-Basisfunktionen: 60, Breite des DGB-Ortsraumintervalls: $0-5 a_0$, Anzahl der numerischen Integrationspunkte: 500, Breite des Integrationsgitters: $-6,+6 a_0$) gemäß den Erfahrungen aus den vorangegangenen Beispielen, bis Sie sich sicher sind, daß zumindest die untersten Eigenwerte und -funktionen hinreichend gut konvergiert sind.
2. **Verständnis der Resultate:** Warum haben die untersten beiden Eigenfunktionen fast (aber nicht exakt) dieselbe Energie? Wie ändert sich diese sogenannte „Tunnelaufspaltung“ mit der Barrierenhöhe bzw. dem Abstand zwischen beiden Potentialtöpfen? Warum heißt diese Aufspaltung „Tunnelaufspaltung“? Warum hat der erste Zustand mit Eigenenergie oberhalb der Barriere ausgerechnet an der Stelle der Barriere ein großes, breites Maximum? Warum treten die Eigenwerte oberhalb der Barriere nicht mehr in Paaren auf?
3. **Absichtliche Lücken im Ortsraum:** Verwenden Sie den zusätzlichen input-Parameter `dg0`, um DGB-Basisfunktionen nur in den Intervallen $[-dg9, -dg0]$ und $[dg0, dg9]$ zu plazieren (das Intervall $[-dg0, +dg0]$ um den Nullpunkt herum bleibt frei von Basisfunktionen). Versuchen Sie, mit Hilfe dieses Tricks möglichst wenig Basisfunktionen in einem möglichst engen Intervall um jeden Potentialtopf herum zu plazieren, aber die beiden niedrigsten Eigenwerte (also das fast-entartete unterste Eigenwertpaar) dennoch auf 5 signifikante Stellen richtig zu berechnen und gleichzeitig die Tunnelaufspaltung zwischen diesen Zuständen qualitativ richtig zu machen. Wieviele Basisfunktionen brauchen Sie dabei insgesamt?
(*Tip:* Verändern Sie dabei sowohl `dg0` und `dg9` als auch die Gesamtanzahl der DGB-Funktionen, da letztere indirekt deren Abstand und Breite bestimmt!)

Experimente mit „double_well_sinus“

Das Programm „double_well_sinus“ ist eine analoge Umstellung des Programms „morse_sinus“ vom Morsepotential auf dasselbe Doppelminimumpotential. Hier ist es nicht so leicht, eine absichtliche Ortsraumücke in den Basisfunktionen zu erzeugen.

Wieviele Sinus-Basisfunktionen brauchen Sie mindestens, um die untersten beiden Eigenwerte im Doppelminimumpotential auf wenigstens 5 signifikante Stellen korrekt zu berechnen, bei qualitativ richtiger Tunnelaufspaltung? Vergleichen Sie mit Ihrem analogen Resultat für die DGB-Basis mit Lücke. Welcher Basissatz ist effizienter?

Gemeinsamkeiten und Unterschiede der vorigen Beispiele zu Methoden der Reaktionsdynamik und Quantenchemie

Alles obige (sowie natürlich noch ausgefeiltere Basissätze, Integrationsmethoden, usw.) verwendet man de facto in der Forschungspraxis zur Berechnung von eindimensionalen Kern-Schwingungswellenfunktionen; d.h. die Prinzipien sind hier zunächst relativ einfach. Problematisch ist der Übergang zu mehr Freiheitsgraden: Im Prinzip kann man höherdimensionale Basisfunktionen und -gitter als sog. „direkte Produkte“ eindimensionaler Basisfunktionen bzw. -gitter erzeugen: Aus einem Gitter von n_1 Punkten in einem Freiheitsgrad und einem Gitter von n_2 Punkten in einem weiteren Freiheitsgrad wird so ein „rechteckiges“ Gitter von $n_1 \times n_2$ Punkten in zwei Dimensionen. Dieses Vorgehen skaliert exponentiell mit der Anzahl der Freiheitsgrade und wird daher auch mit modernen Großrechnern bereits bei etwa 5-atomigen Systemen undurchführbar. Auswege sind Gegenstand aktueller Forschung.

In der Quantenchemie (elektronisches Problem) gibt es einige dieser Ingredienzien ebenfalls praktisch ganz genauso, aber ein paar Aspekte kommen noch hinzu (siehe Vorlesung „TC-I: Quantenchemie“):

- Übergang Einteilchenwellenfunktion (= Orbital) \rightarrow Mehrteilchenwellenfunktion als Hartree-Produkt aus Orbitalen; letztlich sorgt das aber nur für kompliziertere Energieausdrücke und mehr zu rechnende Integrale.
- durch „Antisymmetrisierung“ wird aus dem Hartree-Produkt eine Slaterdeterminante; im Energieausdruck kommt dadurch zum Coulombterm noch der Austauschterm hinzu, aber sonst ist das im Wellenfunktionsbild eigentlich gar keine wesentliche Komplikation.
- Zweiteilchenoperator im Hamiltonoperator \rightarrow Abhängigkeit des „Hamilton“(Fock)-Operators in der Orbitalbasis von seinen eigenen Eigenfunktionen \rightarrow SCF-Zyklus.
- methodische Näherungen: Mittelungen über Teilchen Nr. 2 in Hartree-Fock. Aber auch Näherungen jenseits von Hartree-Fock, also in fast allen Elektronenkorrelationsmethoden (nur full-CI und full-CC sind formal exakt).
Beachte: Hier im Praktikum haben wir immer nur „ein“ Verfahren zur Lösung der Schrödingergleichung, nämlich ein exaktes: Außer der Basisfunktionsentwicklung machen wir *keine* weiteren Näherungen!

5 Zeitabhängigkeit in der Quantenmechanik:

Separation der zeitabhängigen Schrödingergleichung

$$i\hbar \frac{\partial}{\partial t} \Psi(x, t) = -\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} \Psi(x, t) + V(x) \Psi(x, t) =: \hat{H} \Psi(x, t) \quad (73)$$

Wenn Potential V unabhängig von t , Separationsansatz $x \leftrightarrow t$ möglich:

$$\Psi(x, t) = a(t) \psi(x) \quad (74)$$

Einsetzen in Gl. 73 und Division durch $a(t)\psi(x)$:

$$i\hbar \frac{1}{a(t)} \frac{da(t)}{dt} = \frac{1}{\psi(x)} \left(-\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + V(x) \right) \psi(x) \quad (75)$$

Nur erfüllbar, wenn beide Seiten gleich $E = \text{const.}$ sind.

Die linke Seite von Gl. 75 lautet:

$$i\hbar \frac{da(t)}{dt} = E a(t) \quad (76)$$

Das ist eine separierbare Differentialgleichung für $a(t)$:

$$\Rightarrow \frac{da(t)}{a(t)} = -\frac{iE}{\hbar} dt \quad (77)$$

$$\Rightarrow \ln a(t) = -\frac{iEt}{\hbar} + C \quad (78)$$

$$\Rightarrow a(t) = A e^{-iEt/\hbar} \quad (79)$$

Rechte Seite von Gl. 75 = zeitunabhängige Schrödingergleichung:

$$\hat{H} \psi(x) = -\frac{\hbar^2}{2m} \frac{d^2}{dx^2} \psi(x) + V(x) \psi(x) = E \psi(x) \quad (80)$$

Quantisierte Eigenzustände

Die zeitunabhängige Schrödingergleichung für ein Teilchen der Masse m in einer Raumdimension x lautet:

$$\hat{H}\psi(x) := -\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} \psi(x) + V(x)\psi(x) = E_{tot}\psi(x) \quad (81)$$

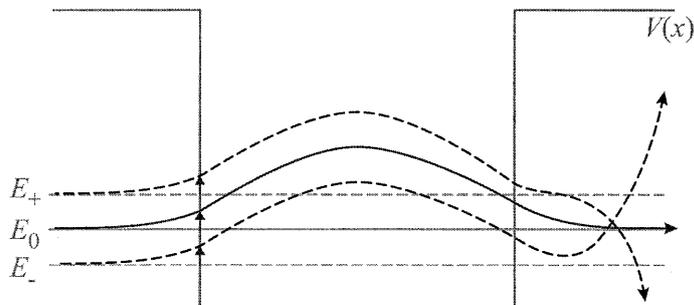
Durch einfache Umformungen ergibt sich:

$$\frac{\partial^2}{\partial x^2} \psi(x) = -\frac{2m}{\hbar^2} [E_{tot} - V(x)]\psi(x) = -\frac{2m}{\hbar^2} E_{kin}\psi(x) \quad (82)$$

⇒ **Konstruktionsvorschrift** für $\psi(x)$ in *beliebigen* eindimensionalen Potentialen $V(x)$:

- 2. Ableitung (Krümmung) von $\psi(x)$ ist proportional zu m , $|E_{kin}| = |E_{tot} - V(x)|$ und $|\psi(x)|$.
- $E_{kin} > 0$: $\psi(x)$ ist zur x -Achse hin gekrümmt,
 $E_{kin} < 0$: $\psi(x)$ ist von der x -Achse weg gekrümmt.
- Wendepunkte von $\psi(x)$ genau dann, wenn $E_{kin} = 0$, also bei $E_{tot} = V(x)$, und bei $\psi(x) = 0$.

⇒ gebundene Zustände sind nur bei *quantisierten* Energien möglich:



Gebundene Zustände im Harmonischen Oszillator

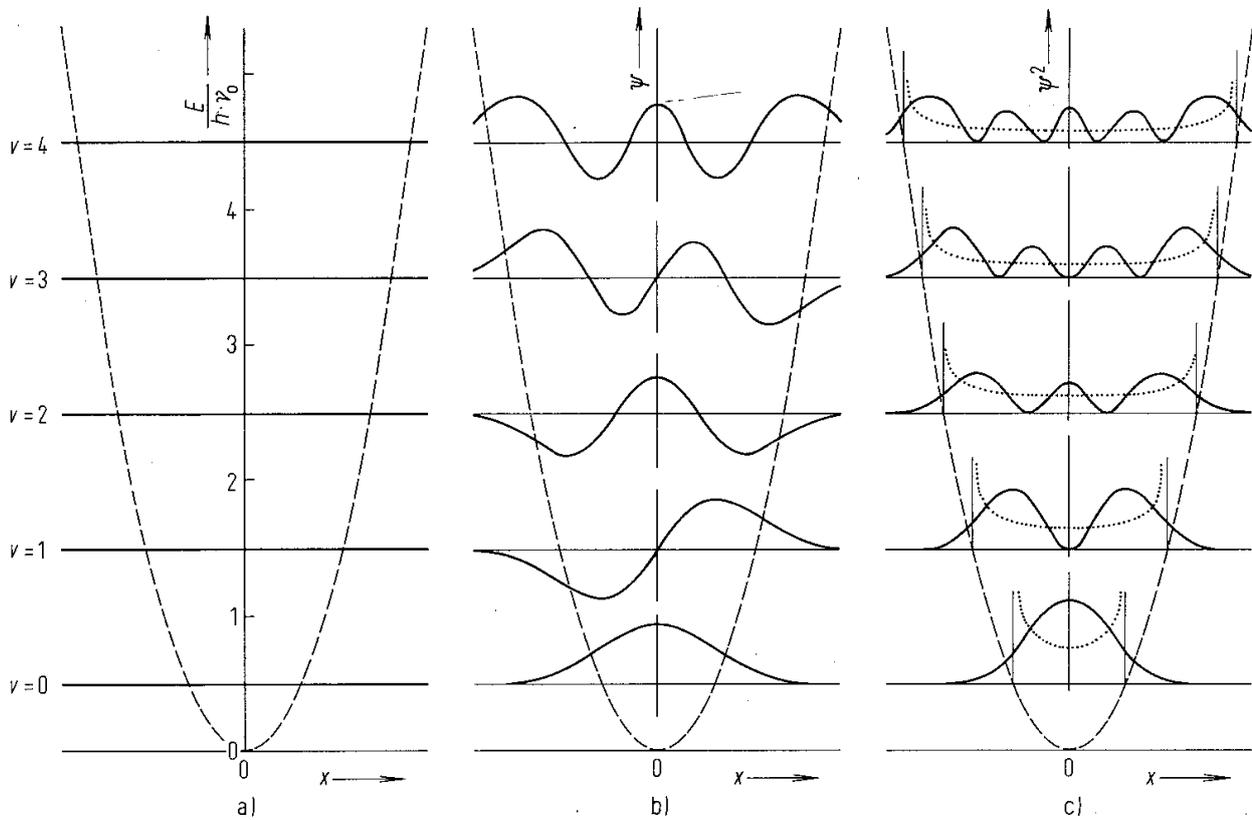


Abb. 3-6. Der harmonische Oszillator
 a) Eigenwerte der Energie
 b) Wellenfunktionen
 c) Wahrscheinlichkeitsdichten.

Die Zeitunabhängigkeit stationärer Zustände

Hat man eine Lösung $\psi(x)$ der zeitunabhängigen Schrödingergleichung gefunden,

$$\hat{H}\psi(x) = E\psi(x) \quad (83)$$

müßte man sie gemäß dem Separationsansatz auf eine Lösung der zeitabhängigen Schrödingergleichung erweitern:

$$\Psi(x, t) = e^{-iEt/\hbar}\psi(x) \quad (84)$$

Diese Zeitabhängigkeit ist jedoch physikalisch folgenlos:

Mittelwerte von Meßwerten sind Erwartungswerte $\langle \Psi | \hat{A} | \Psi \rangle$ von Operatoren \hat{A} :

$$\langle \Psi | \hat{A} | \Psi \rangle = \int_{-\infty}^{\infty} \Psi^* \hat{A} \Psi dx \quad (85)$$

$$= \int_{-\infty}^{\infty} e^{iEt/\hbar} \psi^*(x) \hat{A} e^{-iEt/\hbar} \psi(x) dx \quad (86)$$

$$= e^{i(Et-Et)/\hbar} \int_{-\infty}^{\infty} \psi^*(x) \hat{A} \psi(x) dx \quad (87)$$

$$= \langle \psi | \hat{A} | \psi \rangle \quad (88)$$

Das Resultat ist eine zeitunabhängige Zahl. $\Rightarrow \Psi(x, t)$ bzw. $\psi(x)$ ist ein stationärer Zustand.

In der Chemie geht es aber um *zeitlich veränderliche* Größen!

- Wie können wir nicht-trivial-zeitabhängige Wellenfunktionen konstruieren?
- Wie unterscheiden sie sich von stationären Zuständen?
- Was lernen wir daraus über die Dynamik von Teilchen?

Zeitabhängigkeit durch Superposition

Der Separationsansatz $a(t)\psi(x)$ ist *nicht* allgemein genug!

Die Eigenfunktionen $\psi_n(x)$ des hermiteschen Operators \hat{H} bilden eine vollständige Basis. Also können wir eine allgemeinere Lösung $\Phi(t)$ der zeitabhängigen Schrödingergleichung durch Linearkombination erzeugen:

$$\Phi(x, t) = \sum_n c_n \Psi_n(x, t) = \sum_n c_n e^{-iE_n t/\hbar} \psi_n(x) = \sum_n c_n a_n(t) \psi_n(x) \quad (89)$$

⇒ zeitabhängige Erwartungswerte, z.B. bei $\Phi = a_1\psi_1 + a_2\psi_2$:

$$\langle \Phi | \hat{A} | \Phi \rangle = \langle a_1\psi_1 + a_2\psi_2 | \hat{A} | a_1\psi_1 + a_2\psi_2 \rangle \quad (90)$$

$$= \langle a_1\psi_1 | \hat{A} | a_1\psi_1 \rangle + \langle a_1\psi_1 | \hat{A} | a_2\psi_2 \rangle + \langle a_2\psi_2 | \hat{A} | a_1\psi_1 \rangle + \langle a_2\psi_2 | \hat{A} | a_2\psi_2 \rangle \quad (91)$$

$$= a_1^* a_1 \langle \psi_1 | \hat{A} | \psi_1 \rangle + a_1^* a_2 \langle \psi_1 | \hat{A} | \psi_2 \rangle + a_2^* a_1 \langle \psi_2 | \hat{A} | \psi_1 \rangle + a_2^* a_2 \langle \psi_2 | \hat{A} | \psi_2 \rangle \quad (92)$$

Zwar gilt nach wie vor

$$a_n^* a_n = e^{i(E_n - E_n)t/\hbar} = e^0 = 1 \neq f(t) \quad \text{für } n = 1, 2 \quad (93)$$

aber jetzt haben wir auch Terme mit

$$a_m^* a_n = e^{i(E_m - E_n)t/\hbar} \neq e^0 \quad \text{für } m = 1, n = 2 \quad \text{und} \quad m = 2, n = 1 \quad (94)$$

Also ist $\langle \Phi | \hat{A} | \Phi \rangle$ zeitabhängig.

Beachte:

- die Zeitabhängigkeit wird von den Faktoren $e^{iE_n t/\hbar}$ erzeugt, die bei stationären Zuständen irrelevant waren;
- sind *alle* Eigenfunktionen $\psi(x)$ bekannt und wurden die Entwicklungskoeffizienten c_n zu *einem* (beliebigen) Zeitpunkt bestimmt, kennt man die gesamte Zeitabhängigkeit von Φ .

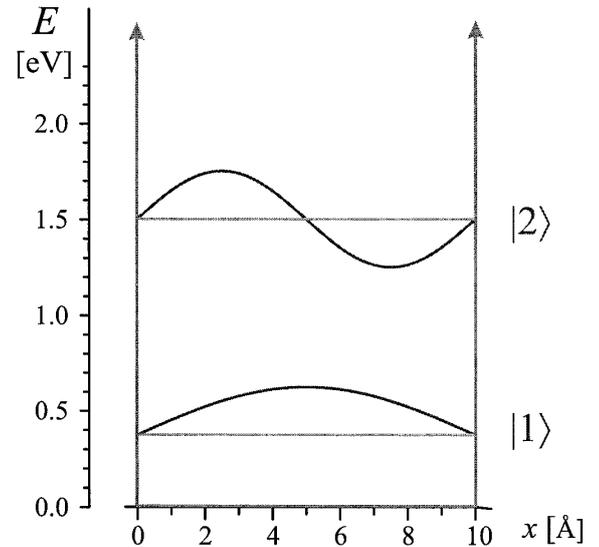
Analytisches Beispiel:

Teilchen im eindimensionalen Kasten (Breite a , unendlich hohe Wände):

Eigenenergien und -funktionen:

$$E_n = \frac{\hbar^2 n^2}{8ma} \quad (95)$$

$$\psi_n(x) = \sqrt{\frac{2}{a}} \sin\left(\frac{n\pi}{a}x\right) \quad (96)$$



1:1-Superposition der beiden niedrigsten Eigenfunktionen (mit ihren Zeit-Vorfaktoren!):

$$\Phi(x, t) = \frac{1}{\sqrt{2}} \{ \Psi_1(x, t) + \Psi_2(x, t) \} \quad (97)$$

$$= \frac{1}{\sqrt{a}} \left\{ e^{-iE_1 t/\hbar} \sin\left(\frac{\pi}{a}x\right) + e^{-iE_2 t/\hbar} \sin\left(\frac{2\pi}{a}x\right) \right\} \quad (98)$$

$$= \frac{1}{\sqrt{a}} e^{-iE_1 t/\hbar} \left\{ \sin\left(\frac{\pi}{a}x\right) + e^{-i(E_2 - E_1)t/\hbar} \sin\left(\frac{2\pi}{a}x\right) \right\} \quad (99)$$

Der Vorfaktor $\exp(-iE_1 t/\hbar)$ ist uninteressant; physikalisch relevant ist nur der Faktor $\exp(-i(E_2 - E_1)t/\hbar)$, der das *relative* Gewicht der beiden Zustände in der Superposition bestimmt.

Dieser Faktor ist zeitabhängig; wir erhalten z.B.:

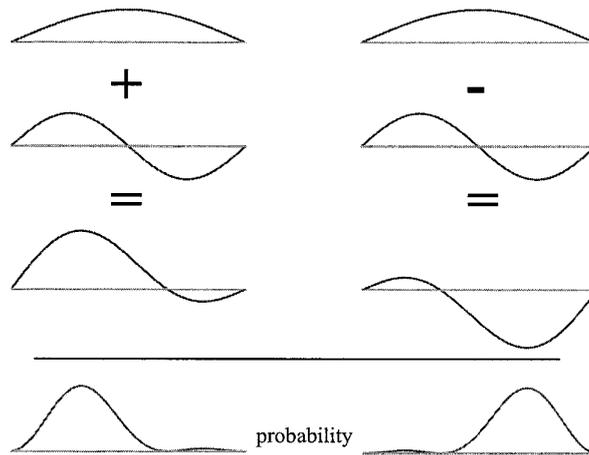
$$\text{für } t = t_0 = 0 : e^{-i(E_2 - E_1)t_0/\hbar} = 1 \quad (100)$$

$$\text{für } t = t_{1/2} = \frac{\pi\hbar}{E_2 - E_1} : e^{-i(E_2 - E_1)t_{1/2}/\hbar} = e^{-i\pi} = -1 \quad (101)$$

Zu diesen Zeiten ergeben sich also diese Superpositionen:

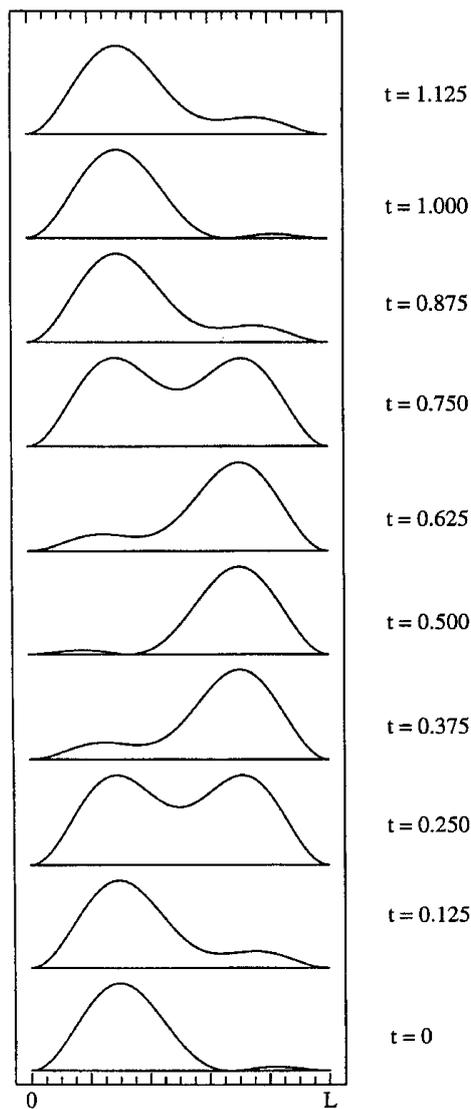
$$t = t_0 : \Psi(x, t_0) = \frac{1}{\sqrt{2}} \{ \psi_1(x) + \psi_2(x) \} \quad (102)$$

$$t = t_{1/2} : \Psi(x, t_{1/2}) = \frac{1}{\sqrt{2}} \{ \psi_1(x) - \psi_2(x) \} \quad (103)$$



Wir erhalten also eine *zeitabhängige* Aufenthaltswahrscheinlichkeitsdichte = *nicht*-stationärer Zustand!

Eine ausführlichere Berechnung ergibt folgenden Zeitverlauf:



Die Energie dieses Zustands ist aus Gl. 92 mit $\hat{A} = \hat{H}$ berechenbar.

Wegen $\hat{H}\psi_n = E_n\psi_n$ und $\langle\psi_n|\psi_m\rangle = \delta_{mn}$ erhalten wir (mit korrekter Normierung):

$$\langle\Phi|\hat{H}|\Phi\rangle = \frac{1}{2}(E_1 + E_2) \quad (104)$$

⇒ jede Einzelmessung liefert

- mit 50% Wahrscheinlichkeit den Wert E_1 , und
- mit 50% Wahrscheinlichkeit den Wert E_2 .

In *diesem* Sinne sind also

- gequantelte Eigenenergien in gebundenen Potentialen die einzig erlaubten (= meßbaren);
- *aber* die zugehörigen stationären Zustände sind *nicht* die einzig erlaubten.

Es gibt zusätzlich eine unendliche Vielfalt weiterer Zustände, die jedoch alle zeitlich veränderlich sind.

Java-applets zur Visualisierung:

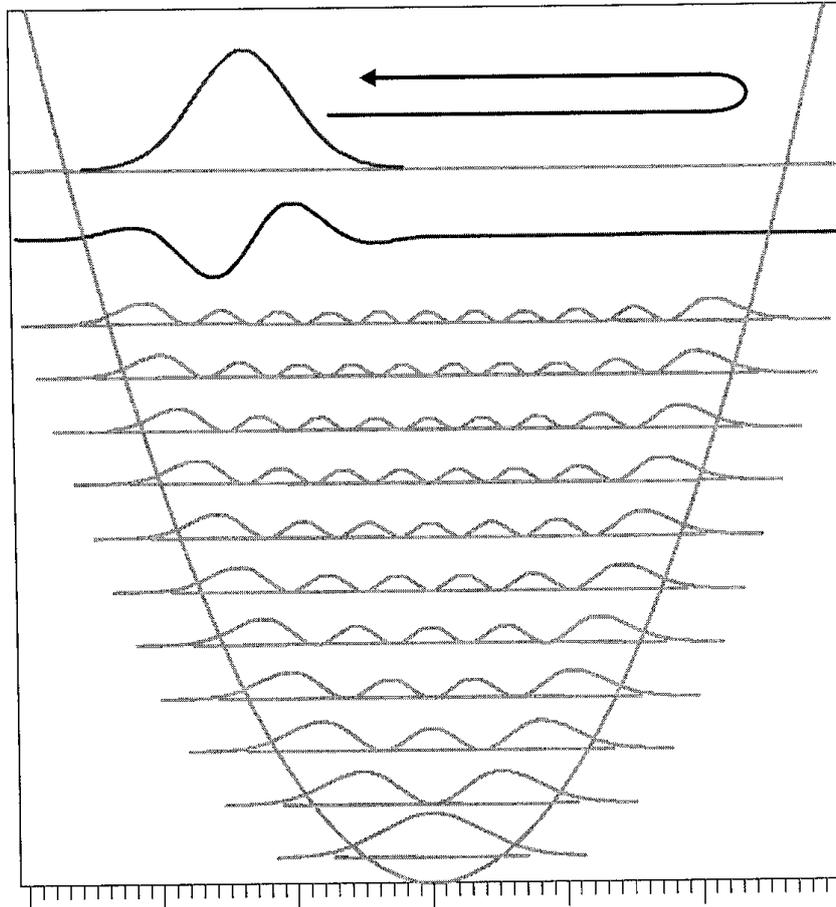
<http://www.quantum-physics.polytechnique.fr/en/index.html>,

dort: → „Quantum superposition in one dimension“ → „Superposition of two stationary states – Another view“

Wellenpaket im Harmonischen Oszillator

Überlagerung von H.O.-Eigenfunktionen mit gaußverteilten Koeffizienten erzeugt ein Gaußpaket mit selber Form wie ($n = 0$)-Grundzustand, aber zeitabhängiger Zentrums-Position. Dieses Gaußpaket bleibt für alle Zeiten gaußförmig; die Position des Zentrums oszilliert mit der Frequenz des klassischen Oszillators.

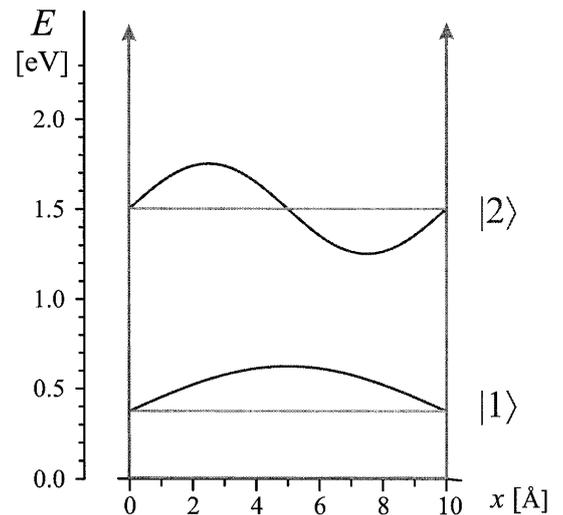
Siehe: <http://www.quantum-physics.polytechnique.fr/en/index.html> → „Quantum superposition in one dimension“ → „Harmonic Oscillator“



Fußnote:

Wie kommt das Teilchen im Kasten im 1. angeregten Zustand von links nach rechts, über den Knoten mit $|\Psi|^2 = 0$ hinweg?

Diese Frage ist falsch gestellt!



- **1. Teilantwort:**

- In stationären Zuständen gibt es keine physikalisch relevante Bewegung.
- Im Gegensatz zur klassischen Mechanik ist $x(t)$ keine zwangsläufig eindeutige Grundgröße.
- Die Frage verlangt eine Antwort der Art: „Zur Zeit t ist das Teilchen sicher am Ort X und nicht am Ort Y“; eine Information dieser Art ist im stationären Zustand $\Psi(x, t)$ aber *nicht* enthalten.

- **2. Teilantwort:**

- In einem geeignet gewählten *nicht*-stationären Zustand ist das Teilchen (nahezu) sicher in der Umgebung des Ortes X und nicht in der Umgebung des Ortes Y – was sich aber mit der Zeit ändert;
- dabei können sich ggf. vorhandene Stellen mit $|\Psi|^2 = 0$ oder $|\Psi|^2 \approx 0$ mitbewegen oder erscheinen und verschwinden;
- in jedem Fall ist das *nicht* mehr der ursprünglich betrachtete stationäre Zustand.

6 Quantenmechanische Dynamik

Propagation von Wellenpaketen mit SPO und FFT

Zur numerischen Lösung der zeitabhängigen Schrödingergleichung

$$i\hbar \frac{\partial}{\partial t} \Psi = \hat{H} \Psi \quad (105)$$

müssen wir numerisch ermitteln können:

- die Wirkung von \hat{H} auf Ψ ; dabei ist der Potentialteil $V\Psi$ trivial, aber der Impulsteil $\hat{T}\Psi$ schwierig;
- die Wirkung von $\frac{\partial}{\partial t}$ auf Ψ .

Dabei stellen wir $\Psi(x, t)$ hier auf äquidistanten Gittern dar, sowohl in x als auch in t . Der Term $V\Psi$ ergibt sich dann einfach durch Multiplikation des Potentials $V(x_i)$ mit der Wellenfunktion $\Psi(x_i)$ an jedem Ortsraum-Gitterpunkt x_i , zum gegebenen Zeitpunkt t_j .

Berechnung der Ortsableitung von Ψ durch FFT:

Für die Berechnung von $\frac{\partial^2}{\partial x^2} \Psi(x, t)$ könnte man finite Differenzen verwenden (s.o.). Damit ist es jedoch schwierig, die nötige hohe Genauigkeit zu erreichen: Da wir zusätzlich in der Zeit iterativ vorwärts propagieren, tendieren auch kleinste Fehler zur Akkumulation \Rightarrow es kommt sehr leicht zu exponentiell ansteigenden Divergenzen.

Abhilfe: Berechnung der Ortsableitung durch Fast-Fourier-Transformation (FFT):

Die Fourier(integral)transformation lautet per Definition:

$$\Psi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \Phi(k) e^{ikx} dk \quad (106)$$

Daher ist die zweite Ableitung im Ortsraum eine einfache Multiplikation mit dem negativen Quadrat der (Impuls-)Koordinate im Impulsraum:

$$\frac{d^2}{dx^2} \Psi(x) = \frac{1}{\sqrt{2\pi}} \frac{d^2}{dx^2} \int_{-\infty}^{\infty} \Phi(k) e^{ikx} dk = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \frac{d^2}{dx^2} \Phi(k) e^{ikx} dk = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} (-k^2) \Phi(k) e^{ikx} dk \quad (107)$$

Also kann man eine (Orts-)Ableitung machen, ohne jemals direkt eine Ableitung zu rechnen – zum Preis von 2 Integraltransformationen. Aufgrund spezieller *fast-Fourier-transform*-(FFT)-Algorithmen kann dieser Weg tatsächlich nicht nur numerisch robuster, sondern auch schneller sein. Die erste und beliebige höhere Ableitungen sind völlig analog berechenbar.

Das gewünschte Resultat $\hat{H}\Psi$ erhält man also durch:

$$\psi(x) \xrightarrow{IFT} \phi(k) \longrightarrow -\frac{\hbar^2 k^2}{2m} \phi(k) \xrightarrow{FT} \frac{\hat{p}^2}{2m} \psi(x) \longrightarrow \frac{\hat{p}^2}{2m} \psi(x) + V(x) \psi(x) = \hat{H} \psi(x) \quad (108)$$

Effiziente FFT-Algorithmen sind nicht-trivial, daher verwendet man in aller Regel fertige FFTs aus subroutinen-Bibliotheken als „black box“ (hier: FFT aus der FFTPACK-Bibliothek, siehe <http://www.netlib.org/fftpack/>). Einige grundlegende Fakten sollte man allerdings kennen:

- Der „natürliche Lebensraum“ der Fouriertransformation ist komplex (s.o.). In speziellen Fällen und mit Tricks kann man auch rein reelle Datensätze mit FFT behandeln. $\Psi(t)$ ist jedoch i.A. ohnehin immer komplex (außer zu ganz speziellen Zeiten t).
- In realer, numerischer Implementation funktioniert FFT nur für diskrete Datensätze endlicher Länge. Daher setzt man bei FFT implizit immer voraus, daß sich der Datensatz vor seinem Anfang und nach seinem Ende periodisch fortsetzt. In diesem Sinn ähnelt FFT eher einer Fourierreihenentwicklung als der analytischen Fouriertransformation.
- FFT beruht letztlich darauf, daß die Transformation eines Datensatzes der Länge n in eine größere Anzahl kürzerer Transformationen der Länge n_i zerlegt wird, mit $n = n_1 \cdot n_2 \cdot n_3 \cdots$. Moderne FFT-Algorithmen funktionieren meist für Datensätze beliebiger Länge. Historisch am ältesten ist FFT für Anzahlen von Punkten, die ein Vielfaches von 2 sind (z.B. 128, 256,...). Dafür sind auch moderne Algorithmen in der Regel noch am schnellsten. Einige funktionieren auch nur für Punktzahlen, die sich in eine begrenzte Anzahl von Primzahlfaktoren zerlegen lassen (z.B. 2, 3 und 5).
- Einige FFT-Implementationen müssen durch einen vorbereitenden Aufruf initialisiert werden. Dies dient u.a. dazu, die obige Primzahlzerlegung durchzuführen.
- Die inverse FFT unterscheidet sich von der FFT „in Vorwärtsrichtung“ nur durch das Vorzeichen im Exponenten des Integralkerns e^{ikx} . Also kann eine FFT-Routine immer beide Transformationsrichtungen, und man muß ihr nur signalisieren, welche der beiden man haben will.
Welche der beiden Richtungen die normale und welches die inverse ist, ist reine Definitionssache. Im vorliegenden Fall werden immer zwei Transformationen entgegengesetzter Richtung hintereinander ausgeführt; dabei ist es egal, welche Transformationsrichtung zuerst kommt.
- Eine N -dimensionale FFT kann in eine Folge $(N - 1)$ -dimensionaler FFTs zerlegt werden. Was hier für den 1D-Fall demonstriert wird, kann also ganz analog auch für größere Systeme mit mehr Freiheitsgraden gemacht werden. In subroutinen-Bibliotheken gibt es in der Regel vorgefertigte Routinen für 1D-, 2D- und 3D-FFT.
- Die maximal mögliche Gitterkonstante im Ortsraum hängt mit dem größten vorkommenden Impulswert zusammen:

$$\Delta x \leq \frac{\pi \hbar}{p_{max}} \quad (109)$$

Das ist eine altbekannte, rein mathematische Eigenschaft der Fouriertransformation. Die berühmte Heisenbergsche Unschärferelation zwischen Ort und Impuls ist letztlich nichts anderes als eben diese Beziehung. Sie kommt zustande, weil in der Quantenmechanik Ort und Impuls über eine Fouriertransformation miteinander verknüpft sind, was wiederum schlicht daran liegt, daß der Impulsoperator als $-i\hbar \frac{d}{dx}$ definiert ist (Ableitung nach der Koordinate), der Ortsoperator aber als Multiplikation mit der Koordinate.

Diskrete Fouriertransformation (DFT) ²

Vereinfachende Annahme: $f(x) = 0$ für $|x| > A/2$. Dann ist die FT:

$$g(\omega) = \int_{-\infty}^{\infty} f(x) e^{i2\pi\omega x} dx = \int_{-A/2}^{A/2} f(x) e^{i2\pi\omega x} dx \quad (110)$$

Mit einer Diskretisierung der x -Achse in $N + 1$ Intervalle der Länge $\Delta x = A/N$ gemäß

$$x_{-N/2} = -A/2, \dots, x_0 = 0, \dots, x_{N/2} = A/2 \quad (111)$$

kann man das Integral Gl. 110 nach der Trapezregel berechnen:

$$g(\omega) \approx \Delta x \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f(x_n) e^{i2\pi\omega x_n} = \frac{A}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f(x_n) e^{i2\pi\omega x_n} \quad (112)$$

Gleichzeitig soll aber auch die inverse FT auf N Punkte diskretisiert werden. A ist größte Wellenlänge, die im Ortsraumgitter darstellbar ist; das entspricht der kleinstmöglichen Frequenz von $\Delta\omega = 1/A$. Also haben wir das Frequenzraumgitter $\omega_k = k\Delta\omega$:

$$\omega_{-N/2} = -\Omega/2, \dots, \omega_0 = 0, \dots, \omega_{N/2} = \Omega/2 \quad (113)$$

mit $\Omega = N\Delta\omega = N/A$. Daraus ergeben sich die wichtigen Beziehungen

$$A\Omega = N \quad , \quad \Delta x \Delta\omega = \frac{1}{N} \quad (114)$$

Mit der Umformung

$$x_n \omega_k = (n\Delta x)(k\Delta\omega) = \frac{nA}{N} \frac{k}{A} = \frac{nk}{N} \quad (115)$$

ergibt sich schließlich:

$$g(\omega_k) = \frac{A}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f(x_n) e^{i2\pi\omega_k x_n} = \frac{A}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f(x_n) e^{i2\pi nk/N} = AF_k \quad (116)$$

Die inverse Transformation lautet:

$$f(x_n) = \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} F_k e^{-i2\pi nk/N} \quad (117)$$

² W. L. Briggs und V. E. Henson: „The DFT: An owner’s manual for the discrete Fourier transform“, Siam, Philadelphia, 1995.

Vorsicht: Traditionell ist die Anordnung im Frequenzraum etwas anders als man zunächst erwarten könnte:

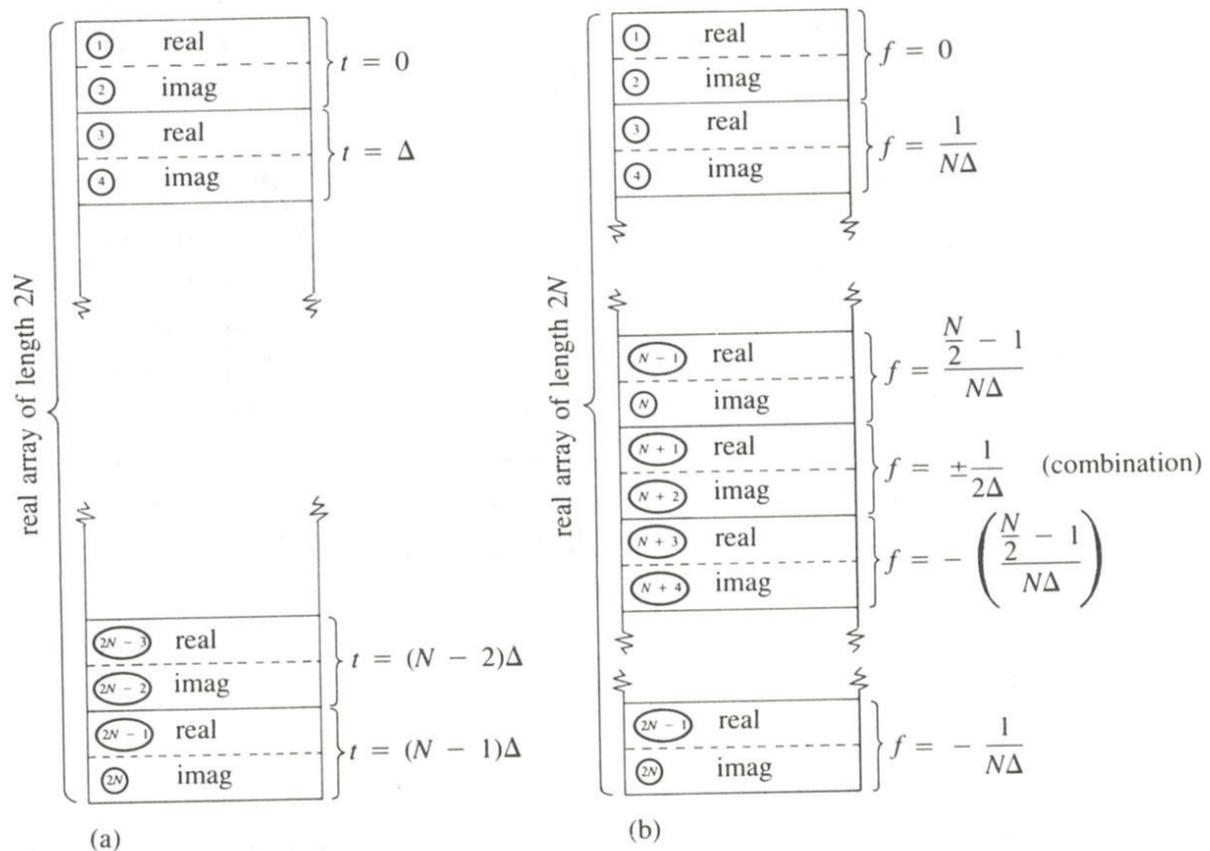


Figure 12.2.2. Input and output arrays for FFT. (a) The input array contains N (a power of 2) complex time samples in a real array of length $2N$, with real and imaginary parts alternating. (b) The output array contains the complex Fourier spectrum at N values of frequency. Real and imaginary parts again alternate. The array starts with zero frequency, works up to the most positive frequency (which is ambiguous with the most negative frequency). Negative frequencies follow, from the second-most negative up to the frequency just below zero.

Vorübung: Einfache Ableitungen mit FFT

Verwenden Sie die zur Verfügung gestellte library FFTPACK, mit folgenden Routinen:

Vorbereitung: call zffti(n,aux)

mit: n: Gitterpunktanzahl, aux: complex(kind=8)-array der Länge $4*n+15$;
genau einmal aufzurufen, *vor* allen anderen FFT-Routinen; berechnet nötige interne
Faktoren für die FFT.

Vorwärtstransformation: call zfftf(n,vector,aux)

mit n, aux wie oben; zusätzlich: vector: complex(kind=8)-array der Länge n;
input: x -Gitterdarstellung der zu transformierenden Funktion $f(x)$,
output: k -Gitterdarstellung der zugehörigen Fouriertransformierten $F(k)$.

Rückwärtstransformation: call zfftb(n,vector,aux)

input: k -Gitterdarstellung der zu transformierenden Funktion $F(k)$,
output: x -Gitterdarstellung der zugehörigen Rücktransformierten $f(x)$.

Technische Falle: Eine Vorwärtstransformation gefolgt von einer Rückwärtstransformation sollte den ursprünglichen Vektor erzeugen. Das ist hier (wie häufig) *nicht* der Fall, weil diese FFT-Routinen die Multiplikation mit dem nötigen Normierungsfaktor $1/n$ "vergessen". Das muß der Benutzer selbst machen, z.B. im k -Raum.

Verwenden Sie den folgenden "magischen" Programmausschnitt zur Erzeugung eines arrays complex(kind=8)::akx1 (bzw. akx2):

```
!build the "magic" k-vector, with the weird ordering FFT needs
twopi=atan(1.d0)*8.d0 ! magic
const=twopi/(dx*dble(n)) ! more magic
nhalfp1=n/2+1 ! still more magic
do i=1,n
  ak=dble(i-1)*const
  if (i>nhalfp1) ak=-(n+1-i)*const ! this is the _really_ weird part
  akx1(i)=(0.d0,1.d0)*ak          ! factor i*k, first derivative
  akx2(i)=-ak*ak                 ! factor (i*k)^2=-k^2, second derivative
!of course, higher derivatives would also work
end do
```

Dabei ist dx der Abstand zweier benachbarter Punkte im äquidistanten Ortsraumgitter.

Im Anschluß können diese akx-arrays verwendet werden, um mit $F(k)$ im k -Raum Element für Element (*ohne* weitere Umordnungen!) multipliziert zu werden, zur Erzeugung der 1. bzw. 2. Ableitung von $f(x)$ (s.o.)

Vorübung: Einfache Ableitungen mit FFT

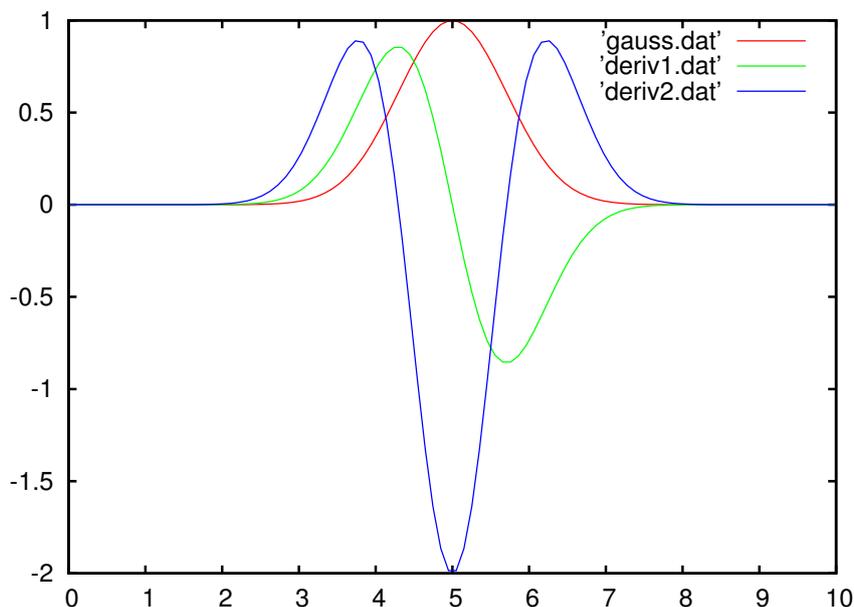
Aufgabe: Erzeugen Sie sich eine nahezu beliebige Funktion $f(x)$ in Gitterdarstellung. Wichtig ist dabei lediglich, daß $f(x)$ ein Stück vor Erreichen des linken bzw. rechten Gitterrands gegen Null geht (sonst entstehen FT-Artefakte, deren Beseitigung hier zu weit führt).

Vorschlag: eine in der Mitte des Intervalls zentrierte Gaußfunktion hinreichend kleiner Breite.

Schreiben Sie ein Programm, das den folgenden Ablauf realisiert:

- durchgängig doppelt genau (komplexe Größen doppelt komplex)
- Eingabe nötiger Parameter, u.a.: Anfang/Ende Ortsraumgitter, Anzahl der Gitterpunkte, ggf. nötige Parameter für $f(x)$
- Schleife:
 - Berechnung aller x -Werte des Ortsraumgitters (für Plotausgaben)
 - Berechnung von $f(x)$ auf allen Gitterpunkten, Abspeichern als Realteil eines komplexen Vektors
 - Ausgabe der $x, f(x)$ -Wertepaare für Plotzwecke
- Aufstellung der `akx`-arrays (siehe Code-Fragment oben)
- Aufruf der FFT-Vorbereitungsroutine
- FFT-“Vorwärts”-Transformation des Funktionsvektors
- Multiplikation mit einem der `akx`-arrays und mit der `real(kind=8)!`-Konstante, die den Wert `1.d0/dble(n)` enthält.
- FFT-“Rückwärts”-Transformation des resultierenden Vektors $\tilde{f}(x)$
- Ausgabe der $x, \tilde{f}(x)$ -Wertepaare für Plotzwecke.

Resultat:



Berechnung der Zeitableitung von Ψ durch SPO:

Für quantitativ exakte Resultate muß auch die Zeitableitung mit aufwendigeren Verfahren berechnet werden. Für unsere Zwecke (und in Verbindung mit der genauen FFT für die Ortsableitung) genügt jedoch eine sehr einfache Methode:

Wenn der Hamiltonoperator \hat{H} explizit zeitunabhängig ist, kann die zeitabhängige Schrödingergleichung

$$i\hbar \frac{\partial}{\partial t} \Psi(t) = \hat{H} \Psi(t) \quad (118)$$

unter Ignorierung der Ortsabhängigkeit von Ψ direkt und einfach integriert werden (wie schon bei der Abseparation der zeitunabhängigen Schrödingergleichung für den Zeitfaktor $f(t)$ vorgeführt), mit dem Resultat:

$$\Psi(t) = e^{-i\hat{H}t/\hbar} \Psi(t=0) \quad (119)$$

Diese Gleichung definiert formal einen neuen Operator, den (Zeit-)Propagator:

$$\Psi(t) = \hat{U}(t, t_0) \Psi(t_0) \quad \text{mit} \quad \hat{U}(t, t_0) = e^{-i\hat{H}(t-t_0)/\hbar} = e^{-i(\hat{T}+\hat{V})(t-t_0)/\hbar} \quad (120)$$

Dieser formal exakte Propagator ist von der Form $\exp(\hat{T} + \hat{V})$, wobei die Operatoren \hat{T} und \hat{V} nicht kommutieren \Rightarrow Eine Trennung dieser Terme in $\exp(\hat{T}) \exp(\hat{V})$ ist nicht exakt (der Fehler enthält den Kommutator $[\hat{T}, \hat{V}]$ als wesentliche Größe), aber trotzdem wünschenswert, da sie oft unterschiedlich behandelt werden (wie auch hier: s.o!).

Einfachste Näherung (mit Abkürzung $\lambda = -i\Delta t/\hbar$): first-order splitting

$$e^{-i\Delta t(\hat{T}+\hat{V})/\hbar} = e^{-i\Delta t\hat{T}/\hbar} e^{-i\Delta t\hat{V}/\hbar} + \frac{1}{2} [\hat{T}, \hat{V}] \lambda^2 + \mathcal{O}(\lambda^3) \quad (121)$$

Besser: second-order splitting

$$e^{-i\Delta t(\hat{T}+\hat{V})/\hbar} = e^{-i\Delta t\hat{V}/(2\hbar)} e^{-i\Delta t\hat{T}/\hbar} e^{-i\Delta t\hat{V}/(2\hbar)} + \frac{1}{24} [\hat{V} + 2\hat{T}, [\hat{V} + \hat{T}]] \lambda^3 + \mathcal{O}(\lambda^4) \quad (122)$$

Aufspaltung höherer Ordnung wurden als effizienter vorgeschlagen, werden aber nicht allgemein verwendet. Das Schema in Gl. 122 (ohne den Fehlerterm mit λ^3) wird abkürzend „split-operator“-Verfahren (SPO) genannt.

Verwendungsstrategie: Jeder Term der Form $\exp(\lambda\hat{O})$ wird in einer Darstellung berechnet, in der er diagonal ist \Rightarrow Bildung der Exponentialfunktion ist dann trivial. Aber: Schnelle Transformationen zwischen den Darstellungen nötig!

Im SPO-Schema sieht ein Zeitschritt also folgendermaßen aus:

- ein halber Zeitschritt mit $\exp(-i\Delta t\hat{V}/(2\hbar))$,
- Fouriertransformation der Wellenfunktion vom Orts- in den Impulsraum,
- ein ganzer Zeitschritt mit $\exp(-i\Delta t\hat{T}/\hbar)$,
- inverse Fouriertransformation vom Impuls- in den Ortsraum,
- ein halber Zeitschritt mit $\exp(-i\Delta t\hat{V}/(2\hbar))$.

Danach liegt die Wellenfunktion einen Zeitschritt Δt später vor. Natürlich kann man genauso gut die Rollen von \hat{T} und \hat{V} vertauschen.

Solange man die Wellenfunktion nicht explizit braucht, kann man auch die halben Zeitschritte mit $\exp(-i\Delta t\hat{V}/(2\hbar))$ am Ende des n -ten und am Anfang des $(n+1)$ -ten Zeitschritts zusammenfassen; dann macht man lediglich alternierend volle Zeitschritte mit $\exp(-i\Delta t\hat{T}/\hbar)$ und $\exp(-i\Delta t\hat{V}/\hbar)$. Die tatsächlich korrekte Wellenfunktion zu einem bestimmten Zeitpunkt erhält man jedoch nur, wenn man vorher und nachher halbe Zeitschritte einfügt.

Norm und (Gesamt-)Energie sollten während der Propagation konstant bleiben (weil die zugehörigen Operatoren $\hat{1}$ und \hat{H} unabhängig von der Zeit sind und mit \hat{H} kommutieren, siehe Vorlesung Theoretische Chemie II). Bei SPO wird die Norm per Konstruktion immer erhalten, die Energie jedoch nicht; also kann man die numerische Genauigkeit der Propagation anhand der Energieerhaltung überprüfen.

Natürlich funktioniert SPO nicht mit beliebig großen Zeitschritten. Zeit und Energie sind durch eine Fouriertransformation miteinander verbunden (wie auch Ort und Impuls), daher gilt hier auch eine ähnliche Beziehung zwischen der Gitterkonstante in der Zeit und dem Maximalwert der Energie. Daraus kann der maximale Zeitschritt abgeschätzt werden:

$$\Delta t \leq \frac{\hbar}{E_{max}} \quad (123)$$

In der Praxis sollte bei schlechten Propagationsschemata der Zeitschritt um ca. einen Faktor 5 kleiner als nach dieser Abschätzung sein. SPO ist recht robust und verträgt größere Zeitschritte.

Für die bequeme Abschätzung von Δx und Δt wird ein kleines Programm `maxi.f90` zur Verfügung gestellt, das anhand der reduzierten Masse des Problems und der maximal möglichen kinetischen und potentiellen Energie die maximal erlaubten Orts- und Zeitschritte berechnet.

Daraus wird klar, daß Vorsicht geboten ist, wenn die potentielle Energie im betrachteten Koordinatenbereich sehr groß wird oder gar gegen unendlich geht \Rightarrow formal unendlich kleine Orts- und Zeitschritte nötig. Das kann man verhindern, indem man das Potential jenseits des chemisch interessanten Bereichs konstant setzt („abschneidet“, `cutoff`).

Aufgabe Quantenmechanische Dynamik

Schreiben Sie ein Programm zur Propagation eines Wellenpakets auf einer eindimensionalen Potentialfläche, unter Verwendung des “split-operator”-Verfahrens (SPO) für die Zeitpropagation und der Fast-Fourier-Transformation (FFT) zur Berechnung der zweiten Ortsableitung der Wellenfunktion. Das modellierte System ist (zunächst) wieder der eindimensionale OH-Oszillator (Parameter s.o.), harmonisch bzw. in Morse-Form.

Zielstruktur des Programms:

- durchgängig doppelte Genauigkeit, auch für komplexe Größen
- Einlesen der nötigen input-Parameter aus einem file:
 - Propagations-Schrittweite `dt` in fs
 - Anzahl an Propagationsschritten (`nt`)
 - Anzahl an Schritten zwischen geschriebenen Zeitschritten (`nframes`)
 - Beginn und Ende des Ortsraumgitters: `x0`, `x9`
 - Anzahl der Punkte im Ortsraumgitter `nx`
 - Reduzierte Masse des OH-Systems in atomic mass units (amu)
 - Potentialparameter `De`, `beta`, `xe`, switch H.O./Morse
 - Cutoffs für das Rechnen und das Plotten
 - Startwellenpaket: Gaußfunktion, dafür Breite `cc` und Ortsraum-Zentrum `psishift`
 - für Plotausgaben: willkürliche Skalierungsfaktoren des Wellenpakets (real, imaginär, quadrat)
 - optional, zu Testzwecken: Stop vor Beginn der Propagation
- Vorbereitung vor der Propagation
 - Einheitenumrechnung der Parameter; Berechnung von Konstanten
 - Berechnung des Potentials $V(\mathbf{x})$ (H.O./Morse) an allen `nx` Integrationspunkten, mit `cutoff`; *Test*: Plotausgabe! (Funktion)
 - Erzeugen Sie die für die FFT notwendigen Vektoren `akx1`, `akx2` (siehe Programmfragment im Skript!)
 - Erzeugung der für die zeitliche Propagation notwendigen Vektoren (*Achtung*: Für die eigenen Variablendefinitionen überprüfen, ob die Vorzeichen stimmen!): `exkin`, `expot`:

$$\mathbf{exkin}(\mathbf{x}) = \exp\left(\frac{i\Delta t}{2m} \mathbf{akx2}(\mathbf{x})\right) \quad (124)$$

$$\mathbf{expot}(\mathbf{x}) = \exp\left(-\frac{i\Delta t}{2} V(x)\right) \quad (125)$$

- Initialisierung der FFT durch Aufruf von `zffti(nx,aux)` (siehe Skript)
- Initialisierung des komplexen Startwellenpakets auf den Gitterpunkten des Ortsraumgitters (evtl Subroutine):

$$\Psi(x) = \exp(-cc(x - \mathbf{psishift})^2) \quad (126)$$

- Test der Norm des Wellenpakets, mit “Banalintegration” und `zabs()` für den Betrag eines komplexen `double` (oder: `sqrt(z*conj(z))` mit `complex(kind=8)::z`):

$$norm = \Delta x \sum_{j=1}^{nx} |\Psi(j)|^2 \quad (127)$$

- Normierung des Startwellenpakets:

$$\Psi(j) = \frac{1}{\sqrt{norm}} \Psi(j) \quad , \quad \text{für alle Gitterpunkte } j \quad (128)$$

- zum Test der Normierung und der Integration: erneute Berechnung und Ausgabe der Norm
- Berechnung und Ausgabe der Energie, mit FFT für die zweiten Ableitung von Ψ und mit “Banalintegration” (Subroutine):

$$\Psi(x) \xrightarrow{\text{zfft}} \Phi(k) \longrightarrow \frac{akx2}{nx} \Phi(k) \xrightarrow{\text{zfft}} \Psi''(x) \quad (129)$$

$$E_{tot} = \Delta x \sum_{j=1}^{nx} \Psi^*(j) \left(-\frac{1}{2m} \Psi''(j) + V(j) \Psi(j) \right) \quad (130)$$

- Ausgabe des Wellenpakets (real- und imaginär-Teil sowie Betrag (s.o.)), geshiftet um die Energie; optionaler Test-Stop
 - Speicherung des normierten Startwellenpakets $\Psi(x, t = 0)$
 - Die Berechnung des Overlaps und der Autokorrelation (s.u.) ist an dieser Stelle noch nicht nötig, dennoch sollten diese aber in entsprechenden Dateien abgespeichert werden.
- eigentliche Propagation, Schleife über `nt` Schritte:

- Die eigentliche Propagation kann im Hauptprogramm oder in einer Subroutine geschehen
- $\Delta t/2$ -Propagation im Ortsraum durch Multiplikation mit dem Propagator `expot`
- FFT in den Impulsraum durch Aufruf von `zfft`
- Δt -Propagation im Impulsraum durch Multiplikation mit dem Propagator `exkin` und Division durch `nx`
(*Denksportaufgabe:* Warum/inwiefern sind die hier durchzuführenden Operationen dieselben bzw. andere als oben bei der Energieberechnung?!)
 - Rück-FFT in den Ortsraum durch Aufruf von `zfft`
 - $\Delta t/2$ -Propagation im Ortsraum durch Multiplikation mit dem Propagator `expot`
 - $t = t + \Delta t$ (auch hier haben wir eine *autonome* DGL, wie in der klassisch-mechanischen Dynamik)
- Ausgabe von Wertepaaren $(t, \text{Overlap}(t))$ und $(t, \text{Autokorellation}(t))$ (letztere komplex und reell), mit “Banalintegration”:

$$overlap = \Delta x \sum_{j=1}^{nx} \Psi^*(j, t) \Psi(j, t) \quad (131)$$

$$autoc = \Delta x \sum_{j=1}^{nx} \Psi^*(j, t = 0) \Psi(j, t) \quad (132)$$

- Alle `nframes` Schritte: Berechnung der Energie und Ausgabe des Wellenpakets (Realteil, Imaginärteil, Betrag), geshiftet um die Energie.

Das Programm „SPO“

Dieses Programm propagiert eine zur Zeit $t = 0$ vorgegebene Anfangswellenfunktion gemäß der zeitabhängigen Schrödingergleichung zu späteren Zeiten, unter Verwendung von FFT und SPO. Die Anfangswellenfunktion ist hier rein reell und hat die Form einer Gaußkurve: $\Psi = b \times \exp(-c \times x^2)$, deren Zentrum an eine vorzugebende Stelle verschoben ist. Dies ist natürlich eine sehr spezielle Wahl; in der Praxis muß die Anfangswellenfunktion sorgfältig an die zu modellierende physikalisch-chemische Situation angepaßt werden.

Als Potentiale sind der OH-Morseoszillator aus „morse_sinus“ und der entsprechende harmonische Oszillator (mit gleicher zweiter Ableitung im Minimum) wählbar.

Struktur des Programms:

- Einlesen von Kontroll-Parametern für das vorliegende System und für die Propagation aus einer vorhandenen Datei `spo.in` (Erklärung s.u.) im aktuellen Verzeichnis.
- Aufruf der subroutine `pot`: Berechnung des Potentials auf allen Orts-Gitterpunkten; Kontrolloutput des Potentials zur graphischen Visualisierung.
- Berechnung einiger Hilfsgrößen für die FFT-Differentiation und die Zeitpropagation.
- Aufruf der subroutine `psinit`: Berechnung einer vorgegebenen Anfangswellenfunktion auf allen Orts-Gitterpunkten; Normierung der Wellenfunktion auf 1.0; Ausgabe der Wellenfunktion.
- Berechnung der Anfangsenergie aus $E = \langle \psi | \hat{H} | \psi \rangle$. Dabei wird die zweite Ableitung im Operator der kinetischen Energie per FFT berechnet, in der subroutine `diffe`.
- In einer großen `do`-Schleife wird dann über `ntot` Zeitschritte zeitlich vorwärts propagiert, mit dem SPO-Schema von Gl. 122. Dabei werden
 - die einfachen Schritte mit dem Potentialoperator direkt im Hauptprogramm ausgeführt,
 - die Schritte mit dem Operator der kinetischen Energie sowie die beiden Fouriertransformationen in der subroutine `diff` gemacht, (überlegen Sie, warum `diffe` und `diff` ähnlich aber nicht identisch sind!)
 - einige Kontrollgrößen berechnet (Norm bzw. „Überlappung der Wellenfunktion mit sich selbst“, Autokorrelation (s.u.), usw.),
 - alle `nframe` Zeitschritte Zeit, Norm und Energie sowie ein „Schnappschuß“ der Wellenfunktion (Realteil, Imaginärteil und Absolutquadrat) ausgegeben.

Erklärung der input-Parameter:

dt: Zeitschritt Δt in Femtosekunden (fs); wird intern in atomic time units (atu) konvertiert;

nt: Anzahl der zu propagierenden Zeitschritte

nframes: nach jeweils soviel Zeitschritten wird ein „Schnappschuß“ der Wellenfunktion ausgegeben;

mass: reduzierte Masse des 1D-Systems; Einheiten: u (bzw. amu);

0/1: Wahl zwischen harmonischem Oszillator und Morseoszillator;

x0,x9: Beginn und Ende des Ortsraumgitters;

nx: Gesamtanzahl der Ortsraumgitterpunkte;

de,beta,xe: übliche Morsepotentialparameter D_e, β, x_e ; Einheiten Å und kJ/mol

cuts: zwei cutoffs für das Potential; in kJ/mol; der erste ist für die tatsächliche Rechnung, der zweite ist für den graphischen output;

cc,psishift: Parameter für die Anfangswellenfunktion, die als Gaußfunktion vorgegeben ist; **cc** ist der Parameter im Exponenten: $\exp(-cc \times x^2)$, also das Inverse der Breite der Gaußfunktion; **psishift** ist eine Verschiebung (in Å) des Zentrums der Gaußfunktion

scale factors: Skalierungsfaktoren, um das Potential und die Wellenfunktionen auf derselben Skala plotten zu können.

0/1: bei Eingabe einer Null in der letzten Zeile hält das Programm nach der ersten Energieberechnung, aber vor der eigentlichen Propagation an (nützlich zu Testzwecken); bei anderen Werten läuft das Programm bis zum Ende durch.

Der output der Wellenpaketschnappschüsse sowie der Potentialoutput sind so gestaltet, daß sie direkt in einer gemeinsame Graphik dargestellt werden können. Für **gnuplot**-Benutzer wird dazu ein script „**movie.gnu**“ zur Verfügung gestellt, das alle Schnappschüsse sukzessive in eine Animation bringt (einfach durch Eingabe von „load 'movie.gnu'“).

Experimente mit „SPO“:

1. **Verständnis des outputs:** Lassen Sie das Programm für den vorgegebenen **harmonischen Oszillator** mit dem vorgegebenen Start-Wellenpaket laufen, dessen Zentrum gegenüber dem Potentialminimum leicht nach rechts verschoben ist. Betrachten Sie die graphische Darstellung des sich in der Zeit bewegenden Wellenpakets, sowie die output-Daten im file `spo.out` (Norm und Energie).
 - a) Kontrollieren Sie Norm- und Energieerhaltung im output.
 - b) Vergleichen Sie die Energie mit den ausgegebenen ersten Eigenenergien des harmonischen Oszillators. Handelt es sich bei der Energie des Wellenpakets um eine „erlaubte“ Eigenenergie? In welchem Sinn ist der vom Programm produzierte Zustand trotzdem „erlaubt“?
 - c) Im Verlauf der Propagation bleibt die generelle Gauß-Form des Wellenpakets beliebig lange erhalten (Höhe und Breite ändern sich allerdings periodisch); dies ist ein sehr ungewöhnliches Artefakt dieser speziellen Potentialform!
 - d) Verifizieren Sie aus den Schnappschüssen, daß das Wellenpaket nach einiger Zeit wieder mit der Form bei $t = 0$ zur Deckung kommt. Im file „`autoc`“ wird die Autokorrelationsfunktion (die Überlappung des Startwellenpakets mit dem laufenden Wellenpaket) als Funktion der Zeit gespeichert. Daraus läßt sich diese „Wiederkehr“ quantitativ und genauer ablesen. Vergleichen Sie die „Wiederkehr“periode mit der klassischen Schwingungsperiode!
 - e) Verändern Sie die Zentrumsposition des Anfangswellenpakets (nicht zuviel!). Verändert sich dabei die Schwingungsperiode des Wellenpakets? Stimmt das mit Ihrer Erwartung überein?
2. **Limits der Propagationsparameter:** Verwenden Sie das Zusatzprogramm „`maxi.f90`“, um für die gegebene reduzierte Masse und die gewünschten Maximalwerte der kinetischen und potentiellen Energie (verwenden Sie hier Werte zwischen dem Potentialcutoff und der tatsächlichen Energie des Wellenpakets) die nach den Gln. 109, 123 maximal möglichen Werte von Δx und Δt zu ermitteln. Machen Sie dann (jeweils ausgehend von den vorgegebenen Parameterwerten als Referenz) folgende Änderungen:
 - a) Verkleinern Sie die Anzahl der Ortsraumgitterpunkte (= Vergrößerung von Δx), bis Sie deutliche Abweichungen von den bisherigen Resultaten sehen (hauptsächlich im Energiewert) und vergleichen Sie mit dem Limit-Wert aus „`maxi.f90`“. Beachten Sie, daß trotzdem noch Norm und Energie erhalten bleiben. Das zeigt, daß der Propagationsalgorithmus sehr robust ist.
 - b) Vergrößern Sie Δt und beobachten Sie die output-Daten. Beachten Sie, daß die Norm immer erhalten bleibt (was daran liegt, daß der Algorithmus per Konstruktion immer unitär ist), und daß sich die Energieerhaltung etwa so verschlechtert, wie von „`maxi.f90`“ vorhergesagt. Was passiert bei noch größeren Zeitschritten?

3. Ersetzen Sie den bisherigen input durch den vorgegebenen input für einen **Morseoszillator**. Verfolgen Sie die völlig unterschiedliche Dynamik des Pakets in diesem Potential.
- a) Warum ist der Koordinatenbereich nach rechts erheblich länger? Was passiert, wenn man den rechten Rand des Koordinatenbereichs ebenso dicht an den rechten klassischen Umkehrpunkt legt wie den linken Rand an den linken Umkehrpunkt?
 - b) Wird das Wellenpaket irgendwann wieder seine ursprüngliche Form annehmen? Experimentieren Sie, aber überlegen Sie auch theoretisch.
 - c) Können Sie trotzdem so etwas wie ein Analogon zur klassischen Schwingungsperiode sehen? Betrachten Sie auch die Autokorrelation! Können Sie mehr als eine Schwingung in der Autokorrelation sehen? Wie ist das zu verstehen?
 - d) Was passiert, wenn die Energie des Pakets oberhalb der Dissoziationsschwelle liegt? Probieren Sie, das Zentrum des Anfangspakets auf den Punkt 0.6 zu setzen. Wie ist das nicht-klassische Resultat zu verstehen, daß in diesem Beispiel ein Teil des Wellenpakets im Potentialtopf gefangen bleibt, obwohl die Gesamtenergie des Pakets deutlich höher ist als die Dissoziationsenergie?

Optionale Experimente für Fortgeschrittene:

Das Programm „spo_mod.f90“ ist eine Variante von „spo.f90“, bei der an 6 Punkten Potentialsprünge eingegeben werden können. Links vom linken und rechts vom rechten Rand wird ein „halber harmonischer Oszillator“ angefügt, der es erlaubt, ein dort geeignet platziertes Anfangswellenpaket eine gewünschte Geschwindigkeit zu geben. Die harmonischen Oszillator-Teile sind weit genug vom Zentrum des Geschehens weg zu plazieren, um es nicht durch Reflektionen zu beeinflussen. Mit Geduld und Experimentierfreude sind damit u.a. folgende Tests möglich (ggf. nach etwas Umprogrammierarbeit):

- Untersuchen Sie das Verhalten eines Wellenpakets an einer **Potentialstufe**. Nach dem Lehrbuch wird folgendes Verhalten erwartet:

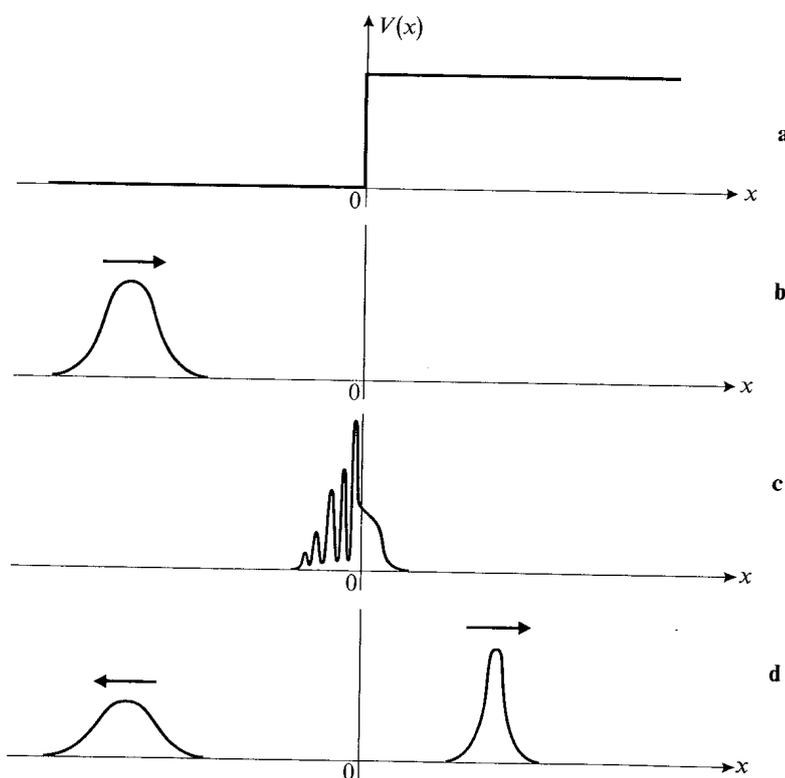
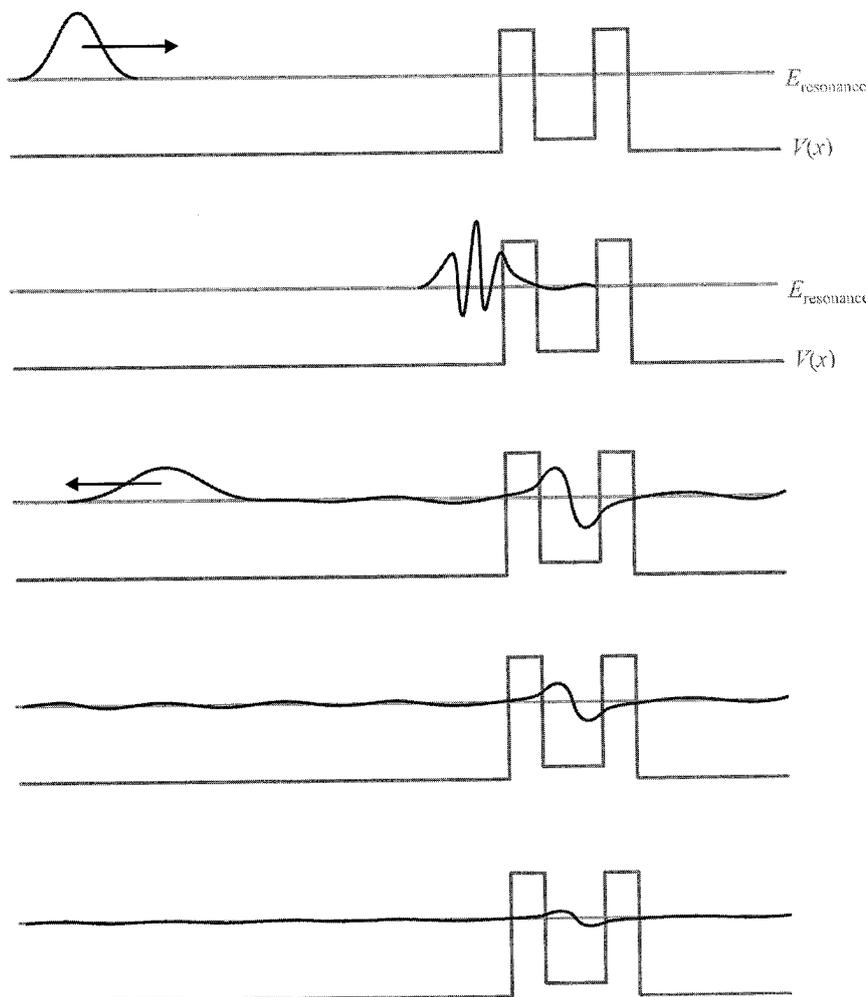


FIGURE 1

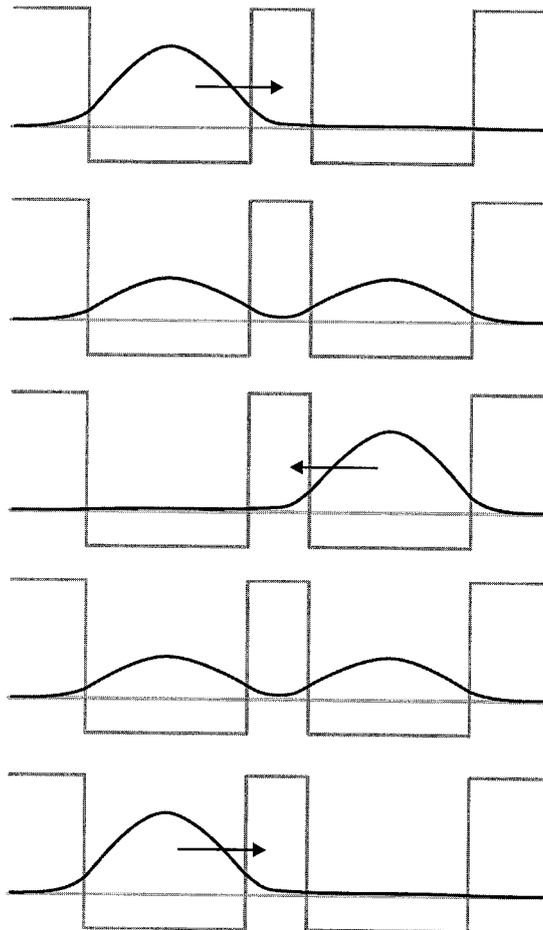
Behavior of a wave packet at a potential step, in the case $E > V_0$. The potential is shown in figure a. In figure b, the wave packet is moving towards the step. Figure c shows the wave packet during the transitory period in which it splits in two. Interference between the incident and reflected waves are responsible for the oscillations of the wave packet in the $x < 0$ region. After a certain time (fig. d), we find two wave packets. The first one (the reflected wave packet) is returning towards the left; its amplitude is smaller than that of the incident wave packet, and its width is the same. The second one (the transmitted wave packet) propagates towards the right; its amplitude is slightly greater than that of the incident wave packet, but it is narrower.

Beobachten Sie, wie sich der reflektierte und transmittierte Anteil mit der Energie des Pakets relativ zur Obergrenze der Potentialstufe ändern (qualitativ im Plot, oder quantitativ durch Integration nur im jeweiligen Raumbereich). Warum gibt es einen transmittierten Anteil, auch wenn die Energie des Pakets unterhalb der Potentialstufe liegt? Warum verschwindet der reflektierte Anteil nicht völlig, wenn die Energie des Pakets über der Potentialstufe liegt? Warum bewegt sich der transmittierte Anteil langsamer als der einlaufende und der reflektierte?

- Ersetzen Sie die Potentialstufe durch eine **Potentialbarriere** endlicher Dicke und Höhe. Wie verändert sich die Tunnelwahrscheinlichkeit durch die Barriere (proportional zur Norm rechts von der Barriere), wenn Sie die Dicke oder die Höhe der Barriere ändern, oder auch die Masse des Teilchens? Wie beeinflusst die Barriere das Wellenpaket, wenn es eine höhere Energie als die Barrierenoberkante hat?
- Setzen Sie eine weiteren Potentialbarriere direkt neben die ersten; dadurch erzeugen Sie eine Doppelbarriere, die auch als **Potentialtopf** mit endlich dicken und endlich hohen Wänden interpretiert werden kann. Dementsprechend gibt es hier nicht nur einen einlaufenden, einen reflektierten und einen transmittierten Wellenpaketanteil: Bei günstiger Wahl der Paketenergie (nämlich gerade dann, wenn sie in der Nähe einer gebundenen Energie des entsprechenden Teilchens im Kasten liegt; diese Energien können Sie mit Lehrbuchformeln berechnen!) bleibt ein Teil des Pakets über eine längere Zeit zwischen der Doppelbarriere gefangen. Dies ist ein quasi-gebundener *Resonanzzustand* mit endlicher Lebensdauer:



- (*Programmierarbeit nötig*) Modifizieren Sie das Programm „spo.f90“ so, daß es das vorher schon verwendete **Doppelminimumpotential** verwendet. Berechnen Sie die Eigenwerte und Eigenfunktionen für dieses Doppelminimumpotential, mit dem stationären Programm (Diagonalisierung der Hamiltonmatrix). Lassen Sie die untersten beiden Eigenzustände ausgeben (Wellenfunktionswerte auf für die Propagation geeigneten Gitterpunkten). Lesen Sie diese beiden Wellenfunktionen in das modifizierte Propagationsprogramm ein und bilden Sie dann eine (komplexe!) Linearkombination dieser Eigenzustände mit den zugehörigen Zeit-Vorfaktoren (wie beim Teilchen im Kasten) und propagieren Sie diesen Superpositionszustand. Erwartetes Verhalten (schematisch; Ihr Potential sieht etwas anders aus):



Wie verändert sich die Oszillationsfrequenz zwischen den beiden Minima mit der Tunnelaufspaltung (= Differenz der beiden untersten Eigenenergien) bzw. mit der Barrierenhöhe, -breite und Teilchenmasse?

Zeitabhängige Spektrensimulation

Ein gegebener Hamiltonoperator habe die Eigenfunktionen $\{\phi_n\}$. Dann kann jede Anfangswellenfunktion $\psi(0)$ in der Basis der Eigenfunktionen ausgedrückt werden:

$$\psi(0) = \sum_n c_n \phi_n \quad \text{mit} \quad c_n = \langle \phi_n | \psi(0) \rangle \quad (133)$$

Die Lösung $\psi(t)$ der zeitabhängige Schrödingergleichung kann dann geschrieben werden als (s.o.):

$$\psi(t) = e^{-i\hat{H}t/\hbar} \psi(0) = \sum_n c_n e^{-iE_n t/\hbar} \phi_n \quad (134)$$

Die *Autokorrelationsfunktion* $A(t)$ von $\psi(0)$ sei definiert durch

$$A(t) = \langle \psi(0) | \psi(t) \rangle = \langle \psi(0) | e^{-i\hat{H}t/\hbar} | \psi(0) \rangle \quad (135)$$

In der Basis der Eigenfunktionen $\{\phi_n\}$ kann $A(t)$ geschrieben werden als:

$$A(t) = \sum_n c_n e^{-iE_n t/\hbar} \langle \psi(0) | \phi_n \rangle = \sum_n |c_n|^2 e^{-iE_n t/\hbar} \quad (136)$$

In dieser Darstellung kann die Fouriertransformierte von $A(t)$ analytisch berechnet werden:

$$S(E) = \frac{1}{T} \text{Re} \int_0^T e^{iEt/\hbar} \langle \psi(0) | \psi(t) \rangle dt \quad (137)$$

$$= \sum_n |c_n|^2 \frac{1}{T} \text{Re} \int_0^T e^{i(E-E_n)t/\hbar} dt \quad (138)$$

$$= \sum_n |c_n|^2 \frac{\hbar}{(E-E_n)T} \sin \frac{(E-E_n)T}{\hbar} \quad (139)$$

$$= \sum_n |c_n|^2 \text{sinc} \frac{(E-E_n)T}{\hbar} \quad (140)$$

Die Funktion $S(E)$ ist also eine Superposition von Funktionen $\text{sinc}(x) = (\sin x)/x$, die je nach dem Wert von E_n gegeneinander verschoben sind. Jede dieser Funktionen $(\sin x)/x$ hat einen „peak“ bei $x = 0$ und an den anderen Stellen kleinere, oszillierende Werte. Im Limit $T \rightarrow \infty$ geht die sinc-Funktion gegen die Delta-Funktion $\delta(x)$.

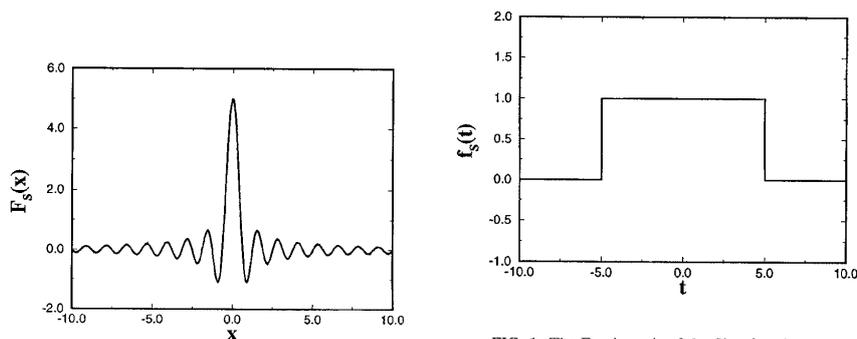
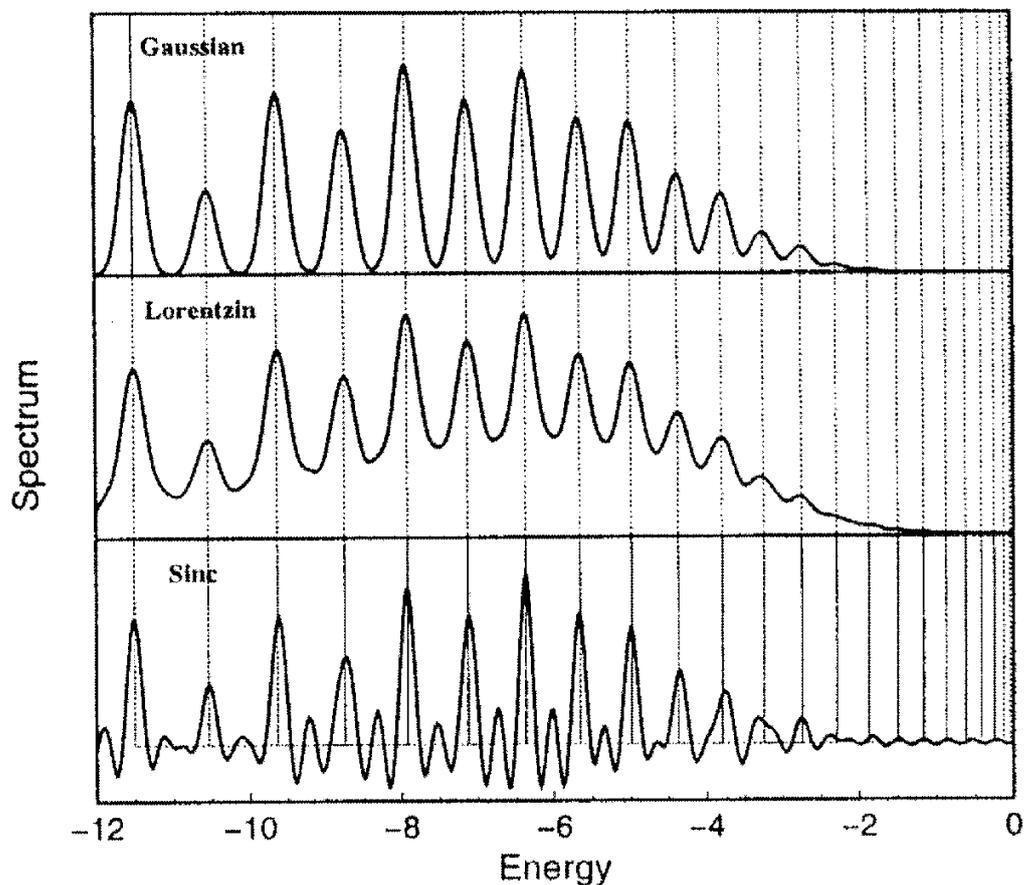


FIG. 1. The Fourier pair of the Sinc function.

$\Rightarrow S(E)$ ist ein *Spektrum* auf der Energie-Achse, mit „peaks“ an den Eigenwerten von \hat{H} . Die Größe von T bestimmt die Auflösung des Spektrums: Bei $T \rightarrow \infty$ ist das Spektrum maximal aufgelöst (*stick spectrum*). Bei finitem aber großem T befindet sich mehr oder weniger „Rauschen“ zwischen den einzelnen Spektralpeaks. Bei finitem, zu kleinem T werden die sinc-Funktionen ggf. so breit, daß einige Spektralpeaks nicht mehr voneinander getrennt erscheinen („Unschärferelation“).

Das Limit T statt ∞ im Fouriertransform-Integral entspricht einer Multiplikation der Autokorrelationsfunktion mit einer Stufenfunktion bis T . Die Fouriertransformierte davon ist gerade die sinc-Funktion.

Verwenden wir stattdessen andere Fouriertransform-Funktionenpaare, z.B. von Lorentz- oder Gaußkurven, ergibt sich ein auch bei kurzen Integrationszeiten rauschfreies Spektrum $S(E)$, bei dem jeder peak einem Eigenwert entspricht; hier gezeigt für die gebundenen Zustände eines Morseoszillators: ³



³ J. Dai und J. Z. H. Zhang, J. Chem. Phys. 103 (1995) 1491.

Beispiele zur Spektrenberechnung:

Im zur Verfügung gestellten Programm `spo.f90` wird bereits die komplexe Autokorrelationsfunktion mitberechnet und ausgegeben (file `autoc_i`). Mit dem ebenfalls fertig zur Verfügung gestellten Programm `falte.f90` kann diese Autokorrelationsfunktion eingelesen und die Zeit-Energie-Fouriertransformation nach Gl. 137 durchgeführt werden (aus technischen Gründen durch eine naive Diskretisierung des Integrals anstatt durch eine „echte“ Fouriertransformation). Als Resultat erhält man ein Spektrum in der Energiedomäne (Realteil des Resultats; der Imaginärteil ist per Konstruktion analytisch Null, siehe Kommentar im Programm).

Technischer Trick: Wie im obigen analytischen Beispiel gezeigt, erhält man starkes „Rauschen“, wenn man dabei die Autokorrelation unverändert integral-transformiert. Das liegt daran, daß die Autokorrelation am Ende des Propagationsintervalls noch nicht auf Null abgeklungen ist, wodurch dort ein artifizierlicher Sprung in der Autokorrelationsfunktion entsteht, der mit in die Fouriertransformation eingeht. Dies wird in `falte.f` dadurch behoben, daß die Autokorrelationsfunktion vor der Integraltransformation mit einer geeigneten Gaußfunktion in der Zeit multipliziert wird, sodaß die modifizierte Autokorrelationsfunktion am Ende des Zeitintervalls tatsächlich Null ist. Dazu muß eine geeignete zeitliche Breite dieser Gaußfunktion angegeben werden: `tmax` in fs. Beachten Sie, daß `tmax` kürzer sein sollte als die Propagationszeit. Kontrollieren Sie das Ergebnis dieser Skalierung durch graphische Darstellung des Absolutbetrags der originalen Autokorrelation (file `autoc` aus `spo.f90`) sowie der skalierten Autokorrelation (file `conv` aus `falte.f90`).

(Randbemerkung: Diese Multiplikation mit einer Gaußfunktion kann entweder als mathematischer Trick verstanden werden oder als Modellierung einer begrenzten energetischen Auflösung in einem experimentellen Spektrometer.)

Berechnen Sie mit `falte.f90` aus Ihren Autokorrelationsfunktionen die zugehörigen Spektren, sowohl für den Harmonischen Oszillator als auch für den Morseoszillator. Der Energiebereich für die Spektren sollte etwa $[0, 1000]$ kJ/mol betragen, etwa 1000 Spektrenpunkte sind empfehlenswert. Verifizieren Sie, daß diese Spektren „peaks“ an den stationären Eigenenergien dieser Potentiale haben. Wie kommen die unterschiedlichen Peakhöhen zustande? (Damit können Sie auch erklären, warum manche Peaks „fehlen“).