

# Computational Physics

## Übungsblatt 3

Miriam Simm

miriam.simm@tu-dortmund.de

Katrin Bolsmann

katrin.bolsmann@tu-dortmund.de

Mario Alex Hollberg

mario-alex.hollberg@tu-dortmund.de

Abgabe: 15. Mai 2020

## Aufgabe 1: Lanczos-Algorithmus

In dieser Aufgabe wird eine fermionische Kette aus  $N$  Gitterplätzen ( $N$  gerade) betrachtet. Der Hamiltonoperator in der Basis  $|i\rangle$ ,  $i \in [1, \dots, N]$  lautet

$$\hat{H} = -t \sum_{i=1}^N (|i\rangle\langle i+1| + |i+1\rangle\langle i|) + \epsilon \left| \frac{N}{2} \right\rangle \left\langle \frac{N}{2} \right| \quad .$$

a)

Der Lanczos-Algorithmus ist implementiert, sodass eine tridiagonale Matrix  $T$  berechnet wird. Mit Hilfe der *Eigen*-Bibliothek werde dann Eigenwerte und Eigenvektoren berechnet. Leider sind die erhaltenen Eigenvektoren und Eigenwerte nicht sinnvoll, was auf eine fehlerhafte Implementierung schließen lässt. Für eine höhere erforderte Genauigkeit ergeben sich immer kleinere Eigenwerte, was darauf schließen lässt, dass der kleinste Eigenwert  $\lambda_{\min} = 0$  ist. Dies ergibt bezogen auf die Aufgabenstellung allerdings unseres Erachtens keinen Sinn, da sich ja ein Elektron auf dem Gitter befindet, die Energie also nicht 0 sein kann. Auch die entsprechenden Eigenvektoren konvergieren gegen den Nullvektor. Die Ergebnisse für Eigenwert und Eigenvektor sind für verschiedene Genauigkeiten (hiermit ist gemeint, was in Z.36 im Programm Aufgabe1.cpp als nicht mehr signifikante Änderung festgelegt wurde) im folgenden aufgeführt:

$$\begin{array}{llll} |\lambda_{\text{then}} - \lambda_{\text{now}}| = 0.01 & \Rightarrow & \lambda_{\min} = 0.398249 & \vec{v}_{\lambda} = \vec{0} \\ |\lambda_{\text{then}} - \lambda_{\text{now}}| = 0.001 & \Rightarrow & \lambda_{\min} = 0.0262954 & \vec{v}_{\lambda} = \vec{0} \\ & & \vdots & \end{array}$$

Leider konnten wir den Fehler in der Implementierung nicht ausfindig machen.

b)

Die Hamilton-Matrix für eine Kette mit sechs Gitterplätzen sieht folglich aus:

$$\hat{H} = \begin{pmatrix} 0 & -t & 0 & 0 & 0 & -t \\ -t & 0 & -t & 0 & 0 & 0 \\ 0 & -t & \epsilon & -t & 0 & 0 \\ 0 & 0 & -t & 0 & -t & 0 \\ 0 & 0 & 0 & -t & 0 & -t \\ -t & 0 & 0 & 0 & -t & 0 \end{pmatrix} \quad \checkmark$$

c)

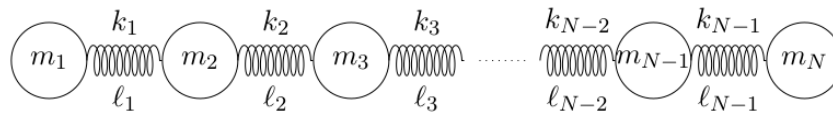
Da der Hamiltonoperator, wie oben ersichtlich, nur wenige von Null verschiedene Einträge hat, ist es sinnvoll, für große  $N$  eine Sparse-Matrix zu verwenden. So werden nur

noch die Einträge ungleich Null gespeichert, wodurch der Speicherbedarf verringert werden kann und eine höhere Effizienz erzielt wird. In Eigen muss dazu "Eigen/Sparse" anstelle von "Eigen/Dense" inkludiert werden.



## Aufgabe 2: Federkette

Gegeben sei eine freie lineare Federkette mit Punktmassen  $m_i$ , Federkonstanten  $k_j$  und Federruhelängen  $l_j$  mit  $i \in \{1, \dots, N\}$  und  $j \in \{1, \dots, N-1\}$ .



Allgemein gilt

$$\begin{aligned} m_1 \ddot{x}_1 &= k_1(x_2 - x_1) \\ m_i \ddot{x}_i &= -k_{i-1}(x_i - x_{i-1}) + k_i(x_{i+1} - x_i) \\ m_N \ddot{x}_N &= -k_{N-1}(x_N - x_{N-1}) \end{aligned}$$

Damit ergibt sich das Differentialgleichungssystem

$$\begin{pmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \vdots \\ \ddot{x}_N \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{-k_1}{m_1} & \frac{k_1}{m_1} & 0 & \dots \\ \frac{k_1}{m_2} & \frac{-k_2-k_1}{m_2} & \frac{k_2}{m_2} & 0 & \dots \\ & & \ddots & & \\ & & & \frac{k_{N-1}}{m_N} & \frac{-k_{N-1}}{m_N} \end{pmatrix}}_{\underline{\underline{A}}} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

Zum Lösen des Differentialgleichungssystems wählen wir den Ansatz  $x_i(t) \sim e^{i\omega t}$ , womit für die Ableitung direkt  $\ddot{\vec{x}} = -\omega^2 \vec{x}$  folgt. Durch Einsetzen von  $\ddot{\vec{x}}$  ergibt sich aus dem Differentialgleichungssystem eine Eigenwertgleichung.

$$\underline{\underline{A}} \vec{x} = -\omega^2 \vec{x}$$

Die Eigenfrequenzen entsprechen also der Wurzel der negativen Eigenwerte. Das Eigenwertproblem wurde in Aufgabe2.cpp für

$$\begin{aligned} m_i &= i \\ k_j &= N - j \\ l_i &= |5 - j| + 1 \end{aligned}$$

gelöst und für die Eigenfrequenzen die Werte

$\omega$
3.95439
2.62228
1.95382
1.51083
1.17586
0.901365
0.663369
$1.4585 \cdot 10^{-9}$
0.44762
0.243446

berechnet. Aufgefallen ist uns, dass die Federruhelängen nicht zur Berechnung der Eigenfrequenzen benötigt wurden. Diese sind also unabhängig von den Anfangsbedingungen.  $\vec{l}$  wird lediglich benötigt, um die explizite Lösung des Differentialgleichungssystems  $\vec{x}$  aufzustellen, diese müssen dann  $\vec{x}(0) = \vec{l}$  erfüllen.

### Aufgabe 3: Integrationsroutinen und eindimensionale Integrale

Ziel der Aufgabe ist die numerische Berechnung der Integrale

$$I_1 = \int_1^{100} dx \frac{e^{-x}}{x} \quad (1)$$

$$I_2 = \int_0^1 dx x \sin\left(\frac{1}{x}\right) \quad (2)$$

Dazu wird die Trapezregel (3), die Mittelpunktsregel (4) und die Simpsonregel (5) implementiert:

$$\int_a^b f(x) dx = h \left( \frac{1}{2}f(a) + f(a+h) + f(a+2h) + \dots + f(b-h) + \frac{1}{2}f(b) \right) \quad (3)$$

$$\int_a^b f(x) dx = h \left( f\left(a + \frac{h}{2}\right) + f\left(a + \frac{3}{2}h\right) + \dots + f\left(b - \frac{3}{2}h\right) + f\left(b - \frac{h}{2}\right) \right) \quad (4)$$

$$\int_a^b f(x) dx = h \left( \frac{1}{3}f(a) + \frac{4}{3}f(a+h) + \frac{2}{3}f(a+2h) + \frac{4}{3}f(a+3h) + \dots + \frac{1}{3}f(b) \right) \quad (5)$$

An die implementierte Funktionsroutine `integration` werden der Intergrand  $f(x)$ , die Intervallgrenzen  $a$  und  $b$ , sowie die Schrittweite  $h$

$$h = \frac{b-a}{n}$$

übergeben. In der untenstehenden Tabelle befinden sich die Ergebnisse der einzelnen Routinen. Die Berechnung endet, wenn eine Genauigkeit von  $10^{-4}$  erreicht ist.

Tabelle 1: Ergebnisse der verschiedenen Integrationsmethoden

	Trapezregel	Mittelpunktsregel	Simpsonregel
$I_1$	0.219386	0.219379	0.219384
$I_2$	0.378534	0.37851	0.378553

# Index der Kommentare

---

3.1      Beziehungsweise beides, wenn wir Dense und Sparse Matrizen verwenden wollen