



seit 1558

Spektralmethoden zur Lösung von Flussgleichungen

BACHELORARBEIT

zur Erlangung des akademischen Grades
Bachelor of Science (B. Sc.)
im Studiengang Physik

FRIEDRICH-SCHILLER-UNIVERSITÄT JENA
PHYSIKALISCH-ASTRONOMISCHE FAKULTÄT
THEORETISCH-PHYSIKALISCHES INSTITUT

eingereicht am 19.09.2014 von Marc Steinhauser
geb. am 18.10.1991 in Biberach (Riß)

Gutachter

Erstgutachter:

Universitätsprofessor Dr. rer. nat. Andreas Wipf
Theoretisch-Physikalisches Institut
Friedrich-Schiller-Universität Jena

Zweitgutachter:

Juniorprofessor Dr. rer. nat. Martin Ammon
Theoretisch-Physikalisches Institut
Friedrich-Schiller-Universität Jena

„MORAL PRINCIPLE 1:

(i) When in doubt, use CHEBYSHEV polynomials unless the solution is spatially periodic, in which case an ordinary FOURIER series is better.

(ii) Unless you're sure another set of basis functions is better, use CHEBYSHEV polynomials.

(iii) Unless you're really, really sure that another set of basis functions is better, use CHEBYSHEV polynomials.“

– JOHN P. BOYD

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	2
2.1	Herleitung der Flussgleichung	2
2.2	Die Flussgleichung des quantenmechanischen anharmonischen Oszillators	5
2.3	Die spektrale Kollokationsmethode	7
3	Implementierung	11
3.1	Herleitungen	11
3.2	Umsetzung	16
4	Vergleich der Methoden	20
5	Zusammenfassung und Ausblick	30
A	Quelltextausschnitte	I
B	Literatur	VIII

1. Einleitung

Zwei universelle Methoden kommen in dieser Arbeit zum Einsatz. Dies ist zum einen die Flussgleichung, deren Eigenschaften hier anhand eines vertrauten Modellproblems – dem quantenmechanischen anharmonischen Oszillator – studiert werden. Zum anderen sind es die Spektralmethoden, welche Differenzialgleichungssysteme sehr effizient lösen können. Durch verschiedene Varianten der Kollokationsmethode wird die Flussgleichung des anharmonischen Oszillators gelöst. Dabei dienen neben den CHEBYSHEV- auch die LEGENDRE-, HERMITE- und LAGUERRE-Polynome als spektrale Basis. Einerseits stehen konventionelle Integratoren, wie das explizite EULER-Verfahren oder das klassische RUNGE-KUTTA-Verfahren mit Schrittweitensteuerung zur Verfügung. Andererseits wurde ein spektraler Integrator geschrieben, der das NEWTON-RAPHSON-Verfahren enthält. Bei ihm kann wiederum zwischen einer Version mit Kompaktifizierung und einer Variante mit Patches gewählt werden.

Zur Formulierung der Flussgleichung wird KENNETH WILSONs Konzept der Renormalisierungsgruppe (RG) benötigt, welche die Skalenabhängigkeit der Physik beschreibt. Makroskopische Phänomene bei großen Längen und kleinen Impulsen werden dabei als mikroskopische Wechselwirkungen aufgefasst. Demnach wirken auch in unserer Alltagswelt die quantenmechanischen Gesetze, deren Auswirkungen wir jedoch nicht bemerken, weil sich die Fluktuationen im Mittel ausgleichen und die resultierenden Effekte extrem schwach ausfallen. Der Grundgedanke der RG besagt, dass im Impulsraum mikroskopische Fluktuationen nicht alle auf einmal zu behandeln sind, sondern nacheinander von Skala zu Skala. Für einen nicht-perturbativen Zugang dienen funktionale Methoden. Durch die Berechnung der erzeugenden Funktionale der Korrelationsfunktionen erhält man die vollständige Information der zugrunde liegenden Theorie, nachdem über alle Fluktuationen integriert wurde. Mit der funktionalen Renormierungsgruppe (fRG) werden lediglich die Änderungen der Korrelationsfunktionen behandelt, die durch infinitesimale Impulsschalenfluktuationen entstehen. Damit ergibt sich eine funktionale Differenzialstruktur, die gegenüber der gewöhnlichen integralen Formulierung eine bessere analytische als auch numerische Zugänglichkeit und Stabilität besitzt.

Wie dieser Absatz verdeutlicht, ist es nützlich, sich mit der funktionalen Renormierungsgruppe und der Flussgleichung zu beschäftigen, da viele Forschungszweige der theoretischen Physik von ihr Gebrauch machen. Das große Interesse an dieser Methode ist darin verwurzelt, dass sie grundsätzlich nicht-perturbativ ist. Oft erzielt die fRG selbst in Fällen, in denen die Störungstheorie scheitert, nützliche Resultate. Beispielsweise kommt die fRG bei der Behandlung des asymptotischen Sicherheitsszenarios in der Quantengravitation zum Einsatz. Darüber hinaus wird die fRG in der statistischen Physik zur Berechnung kritischer Exponenten genutzt. In der Teilchenphysik hilft sie bei der Theorie der starken Wechselwirkung, den elektroschwachen Phasenübergängen und dem Hierarchieproblem. Viele Anwendungen finden sich in der Physik der kondensierten Materie wieder: Sowohl bei der vereinheitlichten Beschreibung klassischer Bosonen, als auch bei der Supraleitung, flüssigem He^4 , Prozessen der Keimbildung & frustrierten Magneten wurde die fRG bereits genutzt. Weitere Einsatzgebiete liegen in der Kernphysik, Atomphysik, Quantenstatistik, Quantenchromodynamik, Hochenergiephysik, Eichtheorie und der Allgemeinen Relativitätstheorie. In den diesem Abschnitt zugrunde liegenden Quellen [1–4] sind noch zusätzliche Anwendungen zu finden.

Zuerst wird der Fokus dieser Arbeit in Kapitel 2 auf die Grundlagen gelegt. Zu Beginn leitet Abschnitt 2.1 die WETTERICH-Gleichung her und im folgenden Abschnitt 2.2 wird aus ihr die numerisch zu berechnende Flussgleichung des quantenmechanischen anharmonischen Oszillators abgeleitet. Als Abschluss der Grundlagen steht in Abschnitt 2.3 die spektrale Kollokationsmethode im Mittelpunkt. Anschließend stellt Kapitel 3 die Implementierung und die dafür notwendigen Formeln vor. Kapitel 4 beinhaltet schließlich alle Resultate sowie deren Vergleiche unter Variation der Eingabeparameter.

2. Grundlagen

Zu Beginn dieses Kapitels wird im Abschnitt 2.1 die WETTERICH-Gleichung basierend auf [1, 5] hergeleitet. Ebenfalls darauf beruhend spezialisiert Abschnitt 2.2 die Flussgleichung für den quantenmechanischen anharmonischen Oszillator. Neben den beiden primären Quellen wurde zusätzlich [6] und in geringem Umfang [3, 7] für den ersten Abschnitt hinzugezogen. Die Grundlagen der genutzten Spektralmethoden sind im letzten Abschnitt 2.3 beinhaltet.

Einfachheitshalber werden in der folgenden Herleitung nur reelle Skalarfelder betrachtet. Alle Rechnungen dieser Arbeit sind in natürlichen Einheiten durchgeführt und Energien sowie Kopplungskonstanten in Einheiten der Eigenfrequenz ω genannt.

2.1. Herleitung der Flussgleichung

Alle n -Punkt Korrelationsfunktionen der Form

$$\langle \phi(x_1) \dots \phi(x_n) \rangle = \mathcal{N} \int \mathcal{D}\phi \phi(x_1) \dots \phi(x_n) e^{-S[\phi]} \quad (2.1)$$

mit der Normierung \mathcal{N} , sodass $\langle \mathbb{1} \rangle = 1$, werden durch das erzeugende Funktional

$$Z[j] = e^{W[j]} = \int \mathcal{D}\phi e^{-S[\phi] + \int d^d y j(y)\phi(y)} \quad (2.2)$$

zusammengefasst und durch dessen funktionale Differenziation

$$\langle \phi(x_1) \dots \phi(x_n) \rangle = \frac{1}{Z[0]} \frac{\delta^n Z[j]}{\delta j(x_1) \dots \delta j(x_n)} \Big|_{j=0} \quad (2.3)$$

berechnet. Das SCHWINGER-Funktional $W[j] = \ln Z[j]$ erzeugt nur zusammenhängende Korrelationsfunktionen und seine LEGENDREtransformierte, die effektive Wirkung

$$\Gamma[\varphi] = \sup_j \left(\int d^d y j(y)\varphi(y) - W[j] \right), \quad (2.4)$$

liefert sogar nur die 1-Teilchen-irreduziblen (1PI). Somit ist sie die effizienteste Variante, um alle Informationen der zugrunde liegenden Quantenfeldtheorie zu bündeln. Denn nichtzusammenhängende FEYNMAN-Diagramme können als Produkt von zusammenhängenden Diagrammen geschrieben werden, die wiederum aus 1PI-Diagrammen konstruierbar sind. In der effektiven Wirkung wird für ein gegebenes φ das zugehörige $j = j_{\text{sup}}$ selektiert und für dieses gilt

$$\frac{\delta}{\delta j(x)} \left(\int d^d y j(y)\varphi(y) - W[j] \right) = 0 \Rightarrow \varphi(x) = \frac{\delta W[j]}{\delta j(x)} = \frac{1}{Z[j]} \frac{\delta Z[j]}{\delta j(x)} = \langle \phi(x) \rangle_j, \quad (2.5)$$

d. h. φ entspricht dem Erwartungswert von ϕ unter Anwesenheit der Quelle j . Aus Gleichung (2.5) lässt sich $j[\varphi]$ bestimmen, welches anschließend in die rechte Seite der Gleichung (2.4) einzusetzen ist. Wird die Funktionalableitung der effektiven Wirkung am Supremum $j = j_{\text{sup}}$ betrachtet

$$\frac{\delta \Gamma[\varphi]}{\delta \varphi(x)} \stackrel{(2.4)}{=} \int d^d y \frac{\delta j(y)}{\delta \varphi(x)} \varphi(y) + j(x) - \int d^d y \frac{\delta W[j]}{\delta j(y)} \frac{\delta j(y)}{\delta \varphi(x)} \stackrel{(2.5)}{=} j(x), \quad (2.6)$$

ergibt sich die Quanten-Bewegungsgleichung. Durch sie „steuert“ die effektive Wirkung unter Berücksichtigung aller Quantenfluktuationen die Erwartungswerte der Felder.

Nun wird die interpolierende Wirkung Γ_k mit einem Impulsschalenparameter k eingeführt. Diese effektive Durchschnittswirkung soll am ultravioletten (UV) Cutoff Λ gegen die nackte Wirkung

S_{nackt} konvergieren sowie für $k \rightarrow 0$ der effektiven Wirkung Γ entsprechen. Um diese zu erzeugen, wird das infrarot (IR) regulierte Funktional

$$Z_k[j] = e^{W_k[j]} = e^{-\Delta S_k[\frac{\delta}{\delta j}]} Z[j] = \int_{\Lambda} \mathcal{D}\phi e^{-S[\phi] - \Delta S_k[\phi] + \int d^d y j(y)\phi(y)} \quad (2.7)$$

eingeführt. Der darin auftretende IR-Regulatorterm lautet im Impuls- bzw. Ortsraum

$$\Delta S_k[\phi] = \frac{1}{2} \int \frac{d^d p}{(2\pi)^d} \phi(-p) R_k(p) \phi(p) \stackrel{(\text{FT})^{-1}}{=} \frac{1}{2} \int d^d x d^d y \phi(x) R_k(x-y) \phi(y). \quad (2.8)$$

Er modifiziert die klassische Wirkung und kann auch als impulsabhängiger Massenterm interpretiert werden. Die Regulatorfunktion $R_k(p)$ muss folgende Eigenschaften erfüllen:

- $\lim_{p^2/k^2 \rightarrow 0} R_k(p) > 0$ für die IR-Regularisierung
- $\lim_{k^2/p^2 \rightarrow 0} R_k(p) = 0 \Rightarrow Z_{k \rightarrow 0}[j] = Z[j], \Gamma_{k \rightarrow 0} = \Gamma$
- $\lim_{k^2 \rightarrow \Lambda \rightarrow \infty} R_k(p) \rightarrow \infty \Rightarrow \Gamma_{k \rightarrow \Lambda} \rightarrow S + \text{const.}$

Häufig wird zusätzlich gefordert, dass für kleine Impulse p die Bedingung $R_k(p) \rightarrow k^2$ erfüllt sein soll. Diese Forderungen führen nicht zu einer eindeutigen Regulatorfunktion. Aus der Vielzahl möglicher Regulatoren wird in Kapitel 2.2 der optimierte Regulator $R_k(p) = (k^2 - p^2) \theta(k^2 - p^2)$ genutzt [8].

Nachdem die Randbedingungen erfüllt sind, kann der Blick auf die dazwischenliegende Trajektorie gerichtet werden. Hierfür wird die interpolierende effektive Wirkung $\Gamma_k[\varphi]$ durch eine leicht abgewandelte LEGENDREtransformation definiert:

$$\Gamma_k[\varphi] = \sup_j \left(\int d^d y j(y)\varphi(y) - W_k[j] \right) - \Delta S_k[\varphi]. \quad (2.9)$$

Infolgedessen ist die effektive Wirkung nicht zwingend konvex. Jedoch wird sie dies für $k \rightarrow 0$, weil in diesem Grenzwert der Zusatzterm $\Delta S_k[\varphi]$ verschwindet. Analog zu oben folgt damit bei $j = j_{\text{sup}}$:

$$\varphi(x) = \langle \phi(x) \rangle_j = \frac{\delta W_k[j]}{\delta j(x)} \quad (2.10)$$

$$\frac{\delta \Gamma_k[\varphi]}{\delta \varphi(x)} = j(x) - \frac{\delta \Delta S_k[\varphi]}{\delta \varphi(x)} \stackrel{(2.8)}{=} j(x) - \int d^d y R_k(x, y) \varphi(y). \quad (2.11)$$

Durch Umformung von Gleichung (2.11) ergibt sich

$$\frac{\delta j(x)}{\delta \varphi(y)} = \frac{\delta^2 \Gamma_k[\varphi]}{\delta \varphi(y) \delta \varphi(x)} + R_k(x, y). \quad (2.12)$$

$\underbrace{\hspace{10em}}_{\equiv \Gamma_k^{(2)}(x, y)}$

Andererseits ist die Funktionalableitung von Gleichung (2.10):

$$\frac{\delta \varphi(y)}{\delta j(z)} = \frac{\delta^2 W_k[j]}{\delta j(z) \delta j(y)} \equiv G_k^{(2)}(y, z). \quad (2.13)$$

Die Kombination der Gleichungen (2.12) & (2.13) führt schließlich zu der wichtigen Identität

$$\delta^{(d)}(x-z) = \frac{\delta j(x)}{\delta j(z)} = \int d^d y \frac{\delta j(x)}{\delta \varphi(y)} \frac{\delta \varphi(y)}{\delta j(z)} = \int d^d y \left(\Gamma_k^{(2)}(x, y) + R_k(x, y) \right) G_k^{(2)}(y, z). \quad (2.14)$$

Folglich ist der Ausdruck in der großen Klammer die Inverse der zusammenhängenden Zwei-Punkt-Funktion $G_k^{(2)}$, die sich in Operatorformulierung schreiben lässt als

$$G_k^{(2)} = \frac{1}{\Gamma_k^{(2)} + R_k} . \quad (2.15)$$

Schließlich ist für die Flussgleichung noch die Ableitung der skalenabhängigen effektiven Wirkung $\Gamma_k[\varphi]$ notwendig, wobei ihr Argument φ festgehalten wird, d. h. k -unabhängig ist. Demzufolge ergibt Gleichung (2.9):

$$\partial_k \Gamma_k[\varphi] = \int d^d y \partial_k j(y) \varphi(y) - \partial_k W_k[j] - \int d^d x \frac{\delta W_k[j]}{\delta j(x)} \partial_k j(x) - \partial_k \Delta S_k[\varphi]. \quad (2.16)$$

Aufgrund von Gleichung (2.10) heben sich bei $j = j_{\max}$ der erste und dritte Summand weg. Zu beachten ist, dass die Differentiation von $W_k[j]$ zwei Beiträge liefert. Der zweite Summand der rechten Seite stammt von der Skalenabhängigkeit des SCHWINGER-Funktional $W_k[j]$ und der dritte Term entsteht durch die k -Abhängigkeit von dessen Argument j . Demnach ist die Ableitung $\partial_k W_k[j]$ so zu verstehen, dass der Parameter und nicht das Argument variiert wird:

$$\begin{aligned} \partial_k W_k[j] &= \partial_k \ln Z_k[j] = \frac{1}{Z_k[j]} \int \mathcal{D}\phi \partial_k (-\Delta S_k[\phi]) e^{-S[\phi] - \Delta S_k[\phi] + \int d^d y j(y)\phi(y)} \\ &\stackrel{(2.8)}{=} -\frac{1}{2} \int d^d x d^d y \partial_k R_k(x, y) \underbrace{\int \mathcal{D}\phi (\phi(x)\phi(y)) e^{-S[\phi] - \Delta S_k[\phi] + \int d^d y j(y)\phi(y)}}_{= \langle \phi(x)\phi(y) \rangle_k} . \end{aligned} \quad (2.17)$$

Mit der Definition der zusammenhängenden Zwei-Punkt-Funktion

$$G_k^{(2)}(x, y) \stackrel{(2.10)}{=} \langle \phi(x)\phi(y) \rangle_k - \varphi(x)\varphi(y) \quad (2.18)$$

folgt für Gleichung (2.16):

$$\begin{aligned} \partial_k \Gamma_k[\varphi] &= \frac{1}{2} \int d^d x d^d y [G_k^{(2)}(x, y) + \varphi(x)\varphi(y)] \partial_k R_k(x, y) - \partial_k \Delta S_k[\varphi] \\ &\stackrel{(2.8)}{=} \frac{1}{2} \int d^d x d^d y G_k^{(2)}(x, y) \partial_k R_k(x, y). \end{aligned} \quad (2.19)$$

Unter Einbeziehung des früheren Resultats (2.15) führt dies zur Flussgleichung

$$\partial_k \Gamma_k[\varphi] = \frac{1}{2} \operatorname{tr} \left(\frac{\partial_k R_k}{\Gamma_k^{(2)}[\varphi] + R_k} \right). \quad (2.20)$$

Diese Gleichung ist eine nichtlineare Funktional-Integro-Differenzial-Gleichung und wird auch geschlossene WETTERICH-Gleichung genannt. Sie ist exakt, da keine Näherungen oder Störungsrechnung angewendet wurden. Obwohl die Herleitung für reelle Skalarfelder vollzogen wurde, ist diese Gleichung für allgemeinere Felder gültig und dient als Ausgangspunkt für viele Zweige der theoretischen Physik. Die darin auftretende Superspur integriert sowohl über kontinuierliche Größen wie Impulse oder Frequenzen, summiert allerdings auch über diskrete Indices. Wie in Abbildungen 1 skizziert, interpoliert die Flussgleichung zwischen der klassischen Wirkung $S_{\text{nackt}}[\varphi]$ und der effektiven Wirkung $\Gamma[\varphi]$, welche die Quantenfluktuationen aller Impulsskalen enthält. Um die WETTERICH-Gleichung anwenden zu können ist eine Trunkierung, d. h. eine Projektion auf einen endlichdimensionalen Unterraum, notwendig. Doch zuvor soll die Flussgleichung für den untersuchten quantenmechanischen anharmonischen Oszillator spezialisiert werden.

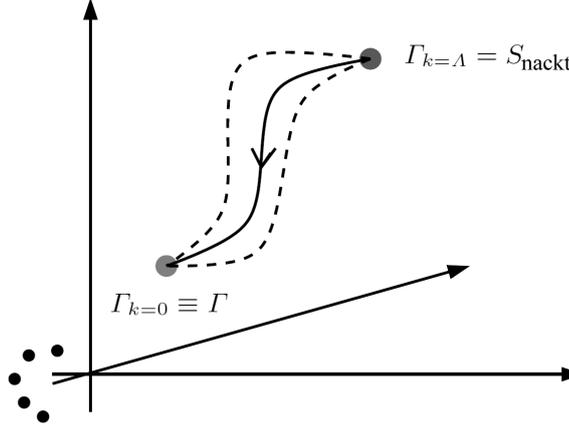


Abb. 1: Skizze des Flusses im Theorienraum. Jede Achse entspricht der Kopplungsstärke eines Operators der effektiven Wirkung, wie z. B. φ^2 , φ^4 oder $\dot{\varphi}^2$. Die Flussgleichung liefert die Trajektorie zwischen der nackten Wirkung $S_{\text{nackt}}[\varphi]$ und der effektiven Wirkung $\Gamma[\varphi]$ (durchgezogene Kurve). Verschiedene Regulatoren R_k führen zu unterschiedlichen Trajektorien (gestrichelte Kurven). In der vollen Theorie ohne Trunkierung ist deren Anfangs- & Endpunkt jedoch fixiert. [5]

2.2. Die Flussgleichung des quantenmechanischen anharmonischen Oszillators

In diesem Abschnitt wird die Flussgleichung des (0+1)-dimensionalen reellen Skalarfeldes, d. h. des EUKLIDischen quantenmechanischen anharmonischen Oszillators, hergeleitet. Ausgangspunkt ist die klassische Wirkung

$$S[q] = \int d\tau \left(\frac{1}{2} \dot{q}^2 + V(q) \right) \quad (2.21)$$

mit dem klassischen Potenzial

$$V(q) = \frac{1}{2} \omega^2 q^2 + \frac{\lambda}{24} q^4, \quad \lambda \geq 0. \quad (2.22)$$

Als Spezialfall ist für $\lambda = 0$ der quantenmechanische harmonische Oszillator enthalten. In der Trunkierung für kleine Energien ist die effektive Wirkung $\Gamma_k[q]$ mit dem skalenabhängigen effektiven Potenzial $u_k(q)$ durch

$$\Gamma_k[q] = \int d\tau \left(\frac{1}{2} \dot{q}^2 + u_k(q) \right) \quad (2.23)$$

gegeben, wobei höhere Ableitungsterme und gemischte Terme der Form $q^n \dot{q}^m$ vernachlässigt werden. Diese Trunkierung wird auch als *local potential approximation* (LPA) bezeichnet.

Unter Annahme $q = \text{const.}$ wird zuerst die Flussgleichung mittels FOURIER-Transformation (FT) $\tilde{R}_k(\tau - \tau') = \int \frac{dp}{2\pi} R_k(p) e^{ip(\tau - \tau')}$ in den Impulsraum transformiert:

$$\begin{aligned} \partial_k \Gamma_k[q] &= \frac{1}{2} \int d\tau d\tau' \frac{\partial_k R_k(\tau - \tau')}{\{\Gamma_k^{(2)}[\varphi] + R_k\}(\tau' - \tau)} \\ &= \frac{1}{2} \int d\tau d\tau' \int_{-\infty}^{\infty} \frac{dp}{2\pi} \frac{dp'}{2\pi} \frac{\partial_k R_k(p) e^{ip(\tau - \tau')} e^{ip'(\tau' - \tau)}}{\{\Gamma_k^{(2)}[\varphi] + R_k\}(p')}. \end{aligned} \quad (2.24)$$

Die Ausführung der τ' -Integration liefert ein $2\pi \delta^{(d)}(p' - p)$ und die anschließende p' -Integration

$$\partial_k \Gamma_k[q] = \frac{1}{2} \int d\tau \int_{-\infty}^{\infty} \frac{dp}{2\pi} \frac{\partial_k R_k(p)}{\{\Gamma_k^{(2)}[\varphi] + R_k\}(p)}. \quad (2.25)$$

Zur FOURIERtransformation von $\Gamma_k^{(2)}(\tau, \tau') = [-\partial_\tau^2 + u_k''(q)] \delta^{(1)}(\tau - \tau')$ ^[1] kann genutzt werden, dass die FT linear ist, q weiterhin konstant gehalten wird und Folgendes gilt:

$$\mathcal{F}[\delta^{(1)}(t)](p) = \int_{-\infty}^{\infty} dt \delta^{(1)}(t) e^{-ipt} = 1 \quad (2.26)$$

$$\mathcal{F}[\partial_t^2 \delta^{(1)}(t)](p) = \int_{-\infty}^{\infty} dt \partial_t^2 \delta^{(1)}(t) e^{-ipt} = \int_{-\infty}^{\infty} dt \partial_t^2 \int_{-\infty}^{\infty} dy e^{iyt} e^{-ipt} = - \int_{-\infty}^{\infty} dy y^2 \delta^{(1)}(y - p) = -p^2. \quad (2.27)$$

Mit der Impulsdarstellung $\Gamma_k^{(2)}(p) = p^2 + u_k''(q)$ folgt anschließend:

$$\partial_k \Gamma_k[q] = \int d\tau \partial_k u_k(q) \stackrel{(2.20)}{=} \frac{1}{2} \int d\tau d\tau' \frac{\partial_k R_k(\tau - \tau')}{\{p^2 + u_k''(q) + R_k\}(\tau' - \tau)}. \quad (2.28)$$

Laut [8] ist der Regulator $R_k(p) = (k^2 - p^2) \theta(k^2 - p^2)$ für das vorliegende Problem die optimale Wahl, weil er die Stabilität der Flussgleichung erhöht. Aufgrund der Annahme, dass q konstant ist, kann die τ -Integration eliminiert werden. Anschließendes Einsetzen von $\partial_k R_k(p) = 2k \theta(k^2 - p^2)$ und p -Integration führt zur Flussgleichung

$$\begin{aligned} \partial_k u_k(q) &= \frac{1}{2} \int_{-\infty}^{\infty} \frac{dp}{2\pi} \frac{2k \theta(k^2 - p^2)}{p^2 + u_k''(q) + (k^2 - p^2) \theta(k^2 - p^2)} = \frac{1}{2} \int_{-k}^k \frac{dp}{\pi} \frac{k}{p^2 + u_k''(q) + k^2 - p^2} \\ &= \frac{1}{\pi} \frac{k^2}{k^2 + u_k''(q)}. \end{aligned} \quad (2.29)$$

Allerdings enthält die Lösung dieser Gleichung noch nicht die korrekte Grundzustandsenergie, wie anhand des freien Teilchens erkennbar ist. Dessen effektives Potenzial $u_\Lambda = 0$ liefert mit Gleichung (2.29) das Resultat $u_k = (k - \Lambda)/\pi$. Um den richtigen Wert $u_k = 0$ zu erhalten, wird die unphysikalische, q -unabhängige Nullpunktenergie subtrahiert:

$$\partial_k u_k(q) = \frac{1}{\pi} \left(\frac{k^2}{k^2 + u_k''(q)} - 1 \right) = -\frac{1}{\pi} \frac{u_k''(q)}{k^2 + u_k''(q)}. \quad (2.30)$$

Diese nichtlineare partielle Differenzialgleichung kann wie in [1] mit einem Polynomansatz oder wie in dieser Arbeit mit einem Spektralansatz (siehe Kapitel 2.3) approximativ gelöst werden. In beiden Fällen wird die Flussgleichung auf ein gekoppeltes Differenzialgleichungssystem reduziert. Dabei sollte ausgenutzt werden, dass durch den Ansatz eines geraden klassischen Potenzials in Gleichung (2.22) die rechte Seite der Flussgleichung (2.30) am Cutoff ebenfalls gerade ist und deren Lösung dadurch für alle Skalen k gerade bleibt. In Abbildungen 2 ist links der Fluss des harmonischen Potenzials gezeigt.

Durch die Wahl $\omega^2 = -1$ besitzt das Potenzial im Ursprung ein lokales Maximum und bei $q = \pm\sqrt{6/\lambda}$ globale Minima. Aufgrund der Tatsache, dass der Nenner $k^2 + u_k''(q)$ bei großen Skalen k positiv ist und der Form der Flussgleichung (2.29) ist der Nenner für alle k positiv. Die größte Veränderung erfährt das effektive Potenzial $u_k(q)$ deshalb dort, wo die zweite Ableitung $u_k(q)''$ minimal ist – im Ursprung. Aus der Gleichung (2.30) ist schließlich zu erkennen, dass das effektive Potenzial entlang kleiner werdendem k zunimmt (abnimmt), wenn seine zweite Ableitung $u_k''(q)$ positiv (negativ) ist. Als Resultat entsteht dadurch im IR ein konvexes Potenzial. Abbildung 2 belegt diese Überlegungen in ihrem rechten Plot anschaulich anhand einer Lösung des anharmonischen Potenzials.

^[1] $\delta^{(d)}(y)$ bezeichnet die d -dimensionale Deltadistribution, nicht ihre d -fache Ableitung. Die d -fache Differenziation wird mit ∂_y^d gekennzeichnet.

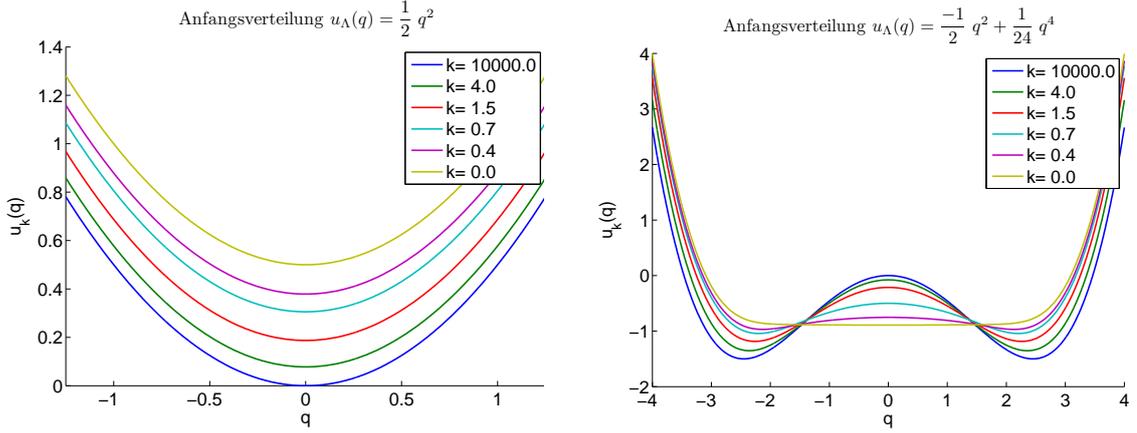


Abb. 2: Fluss des Potentials vom Cutoff $\Lambda = 10000$ bis $k = 0$. Links entspricht das klassische Potenzial $V(q) = u_{k=0}(q)$ dem harmonischen Oszillator und rechts dem anharmonischen Oszillator mit Potenzialbarriere. Beide Darstellungen zeigen die Ergebnisse des *ode45*-Lösers, links mit $Nq = 5$, $q_{max} = 1,25$, $lambda = 0$ & $\omega^2 = 1$ und rechts mit $Nq = 8$, $q_{max} = 4$, $lambda = 1$ & $\omega^2 = -1$.

Zur Bestimmung der Energie dient bei positivem $\omega = \omega_{k=0}$ die Tatsache, dass das effektive Potenzial $u_k(q)$ stets im Ursprung minimal ist. Demnach beträgt die Grundzustandsenergie

$$E_0 = \min(u_{k=0}(q)) = u_{k=0}(0). \quad (2.31)$$

Die Energie des ersten angeregten Zustandes setzt sich aus der Grundzustandsenergie und der Krümmung am Minimum zusammen:

$$E_1 = E_0 + \sqrt{u''_{k=0}(0)}. \quad (2.32)$$

Auch für $\omega^2 < 0$ lassen sich diese beiden Formeln (2.31) & (2.32) ohne Anpassung weiterhin nutzen. Denn das Potenzialminimum liegt im anharmonischen Fall für $k=0$ aufgrund der Konvexität ebenfalls im Ursprung, wie Abbildungen 2 belegt.

2.3. Die spektrale Kollokationsmethode

Die Grundlagen dieses Abschnitts basieren hauptsächlich auf [9]. Des Weiteren beruht er teilweise auf den Quellen [10, 11]. Selbst hergeleitet wurden hingegen die Differenzationsmatrizen \mathbf{D}_q der Gleichungen (3.17) & (3.18).

Spektralmethoden beruhen auf der Grundidee, dass die gesuchte Funktion $u(x)$ durch eine Summe von $N+1$ Basisfunktionen $B_n(x)$ approximierbar ist:

$$u(x) \approx u_N(x) = \sum_{n=0}^N a_n B_n(x). \quad (2.33)$$

Ist L der Operator der Differentialgleichung $Lu = f(x)$, so führt das Einsetzen des Lösungsansatzes (2.33) auf die Residuumsfunktion

$$R(x; a_0, a_1, \dots, a_N) = \|Lu_N - f\|, \quad (2.34)$$

die für die exakte Lösung verschwindet. Folglich ist das Ziel die Residuumsfunktion zu minimieren, um die bestmögliche Näherungslösung zu erhalten. Bei der in dieser Arbeit genutzten

Kollokationsmethode werden die Koeffizienten a_n so bestimmt, dass das Residuum R an den zur Basis $B_n(x)$ gehörenden Stützstellen x_i verschwindet. Dadurch wird die Differenzialgleichung an den sogenannten Kollokationspunkten exakt erfüllt und mit wachsendem N an den dazwischenliegenden x -Werten immer genauer gelöst. Für jede Basis existieren ausgezeichnete Kollokationspunkte x_i , welche die spektrale Konvergenz gewährleisten. Diese entsprechen meist den Nullstellen oder Extrema der zugrunde liegenden Basispolynome (vgl. Gleichungen (3.7) bis (3.10)).

Bei der Wahl geeigneter Basisfunktionen $B_n(x)$ müssen die Kriterien der einfachen Berechenbarkeit, schnellen Konvergenz und Vollständigkeit der Basis erfüllt sein, damit bei ausreichend großen N jede Lösung mit beliebiger Genauigkeit darstellbar ist. Dafür ist eine gewöhnliche FOURIER-Reihe meist am besten geeignet. Inbegriffen ist darin auch die CHEBYSHEV-Reihe $T_n(x)$, die nichts weiteres als eine getarnte FOURIER-Kosinus-Reihe erzeugt und sich mit der simplen Koordinatentransformation $x = \cos(y)$ zu jener transformiert. Dadurch ist die CHEBYSHEV-Reihe automatisch 2π -periodisch in y und besitzt auch für nichtperiodische Funktionen $f(x)$ spektrale Konvergenz. Wegen der Symmetrie von $\cos(y)$ muss auch $f(\cos(y))$ eine gerade Funktion sein, weshalb in der FOURIERdarstellung die Kosinus-Terme ausreichen. In Abbildungen 3 soll die Substitution veranschaulicht werden.

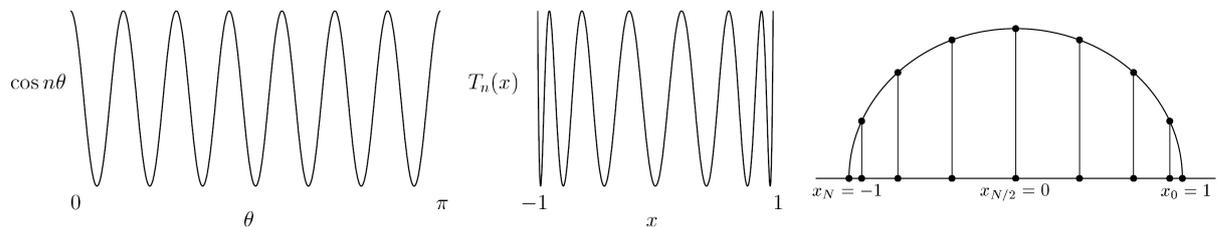


Abb. 3: Das CHEBYSHEV-Polynom $T_n(x)$ kann als Kosinus (links) angesehen werden, der um einen Zylinder gewickelt und von der Seite betrachtet wird (Mitte). Rechts ist die graphische Konstruktion der ungleich verteilten CHEBYSHEV-Stützstellen gezeigt. Dazu wird der Halbkreis mit Einheitsradius in gleichmäßige Stücke unterteilt. Die so erhaltenen FOURIER-Stützstellen werden auf die Grundlinie projiziert. [10]

Ein kurzer Vergleich mit den Finite-Elemente-Methoden (FEM) und den Finite-Differenzen-Methoden (FDM) soll die Vorteile der Spektralmethoden (SM) verdeutlichen. Finite-Elemente-Methoden besitzen einen ähnlichen Grundgedanken wie die Spektralmethoden. Im Gegensatz zu ihnen unterteilen sie das Intervall jedoch in Teilintervalle und setzen auf diesen lokale Basispolynome mit festem Grad an, die nur auf einer kleinen Anzahl Teilintervalle nichtverschwindend sind. Konträr dazu nutzt die Spektralmethode globale (z. T. trigonometrische) Basispolynome hohen Grades, die (abgesehen von ihren Nullstellen) auf dem ganzen Intervall ungleich null sind. Finite-Differenzen-Methoden verwenden hingegen eine Folge überlappender Polynome niedrigen Grades, die $u(x)$ an den Gitterpunkten interpolieren. Die Ableitung $u'(x)$ wird durch die Ableitung der lokalen Interpolation genähert, was zu einer gewichteten Summe der Funktionswerte an den Stützstellen führt.

Obwohl Spektralmethoden vollbesetzte Matrizen erzeugen, sind sie effizienter als die FEM oder FDM, weil ihre Basisfunktionen hohen Grades für ein gegebenes N eine hohe Genauigkeit liefern. Auf zwei Weisen profitiert die Spektralmethode von einem größer werdendem N . Zum einen nimmt die Gitterweite h zwischen zwei Stützstellen ab, zum anderen erhöht sich der Grad der Basispolynome. Daraus folgt die (etwas optimistische) Ordnung des spektralen Fehlers $\mathcal{O}(N^{-N})$, die als exponentielle Konvergenz bekannt ist.

Aus Sicht der FOURIER-Reihe kann die Herkunft des Fehlers auch folgendermaßen begründet werden: Spektralmethoden sind für glatte Funktionen derart genau und effektiv, da ihre FOURIERtransformierten schnell abfallen. Aufgrund ihrer geringen Veränderungen treten im

FOURIERRaum nur die kleinen Moden auf. Folglich sind hohe Wellenzahlen, die mit schnellen Oszillationen korrespondieren, nicht vorhanden. Somit sind auch die von der Diskretisierung stammenden Aliasing-Fehler gering. Dadurch ist für analytische Funktionen in der Regel asymptotisch (also $n \gg 1$) spektrale Konvergenz

$$a_n \sim \mathcal{O}[\exp(-qn^r)], \quad q = \text{const.} > 0, \quad r > 0 \quad (2.35)$$

gegeben. Verfeinerte Kategorien exponentieller Konvergenz sind durch

$$\lim_{n \rightarrow \infty} \frac{\log |a_n|}{n} = \begin{cases} \infty & \text{supergeometrisch} \\ \text{const.} & \text{geometrisch} \\ 0 & \text{subgeometrisch} \end{cases} \quad (2.36)$$

definiert. Grafisch sind die Konvergenzkategorien, wie in Abbildungen 4 zu erkennen ist, leicht unterscheidbar. Im halblogarithmischen Plot liegen die geometrisch konvergierenden Koeffizienten auf einer geraden Linie, wohingegen es in der doppelt logarithmischen Darstellung die algebraisch abfallende Koeffizienten sind.

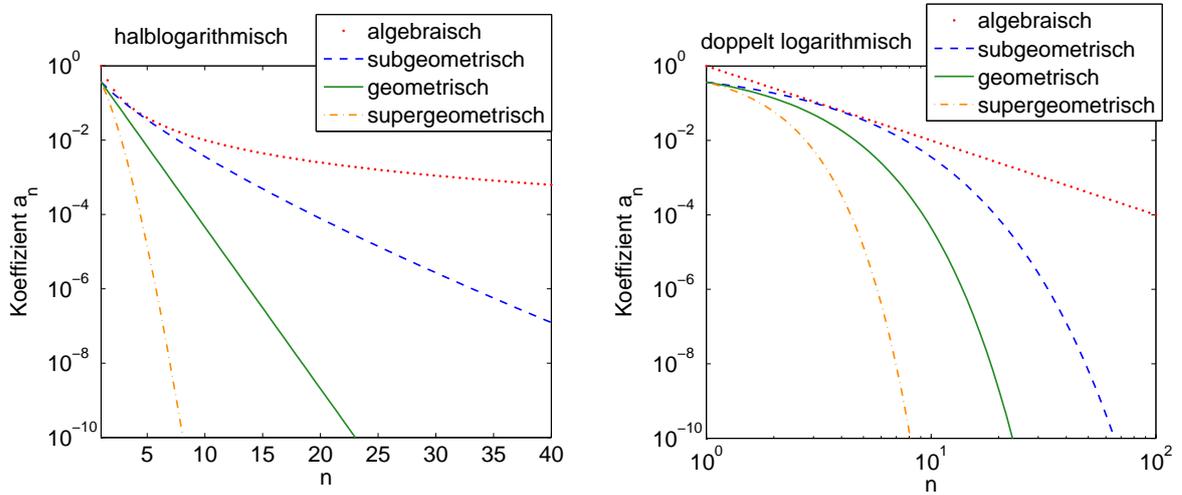


Abb. 4: Abfall der Koeffizienten für die vier verschiedenen Konvergenzraten algebraisch ($a_n \sim 1/n^2$), subgeometrisch ($a_n \sim \exp[-n^{3/4}]$), geometrisch ($a_n \sim \exp[-n]$) und supergeometrisch ($a_n \sim \exp[-n^{3/2}]$). Aufgetragen ist der Betrag des Koeffizienten a_n über dem Koeffizientenindex n – links: halblogarithmisch, rechts: doppelt logarithmisch.

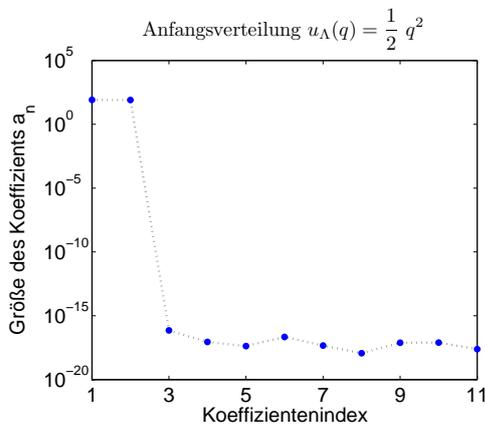


Abb. 5: Auftreten des Rundungsplateaus bei den Parametern $\lambda=0$ & $q_{\max}=18$.

Unabhängig von der Art der Konvergenz endet der Koeffizientenabfall im Bereich der Maschinengenauigkeit im sogenannten Rundungsplateau. Bezeichnet $a_{\max} = \max_n a_n$ und $\varepsilon \approx 10^{-14} \dots 10^{-16}$, so entstehen rundungsfehlerbehaftete Koeffizienten, sobald die exakten Koeffizienten unter εa_{\max} abfallen. In der Konsequenz knickt die Kurve dort ab und alle folgenden Koeffizienten schwanken zufällig um den Wert εa_{\max} , wie Abbildungen 5 verdeutlicht.

Obwohl bisher nur die Koeffizienten betrachtet wurden, lassen sich die Resultate auch auf Funktionswerte übertragen, sofern die Basispolynome beschränkt sind. Dann liegen diese in der Größenordnung $\mathcal{O}(1)$ und die Ordnung des Terms $a_n B_n(x)$ stimmt mit derjenigen von a_n überein. Allerdings ist dies nur bei der

CHEBYSHEV- und LEGENDRE-Basis erfüllt, weil die HERMITE- und LAGUERRE-Polynome in ihrem unendlichen Definitionsgebiet unbeschränkt anwachsen.

Drei verschiedene Fehlerquellen entstehen bei den Spektralmethoden. Zum einen tritt der numerische Trunkierungsfehler^[2] $E_T(N)$ auf, der aus der Vernachlässigung aller Koeffizienten a_n mit $n > N$ resultiert. Zum anderen weicht die Lösungsfunktion, welche durch eine (Pseudo-) Spektralmethode mit $N+1$ Basispolynomen berechnet wurde, von der exakten Lösung ab. Die Differenz zwischen den ersten $N+1$ Termen dieser numerischen Lösung und den entsprechenden Termen der exakten Lösung führen zum Diskretisierungsfehler $E_D(N)$. Zuletzt entsteht der Interpolationsfehler $E_I(N)$ aufgrund der Interpolation durch ein Polynom des Grades $N+1$, das auf den Funktionswerten an den $N+1$ Kollokationspunkten basiert. Darin inbegriffen ist auch der Fehler der in dieser Arbeit genutzten Pseudospektralmethode, die Interpolation enthält. Im Allgemeinen lassen sich diese Fehler der unbekanntenen Lösung nicht bestimmen, allerdings kann die im Folgenden genannte Faustregel genutzt werden. Laut dieser besitzen alle drei Fehler die gleiche Größenordnung, wodurch die Effektivität verschiedener Algorithmen und die notwendige Trunkierung N für eine gewünschte Genauigkeit nur durch Betrachtung des Trunkierungsfehlers $E_T(N)$ abschätzbar sind. CHEBYSHEVs Trunkierungstheorem besagt, dass der numerische Trunkierungsfehler durch die Summe der Beträge aller vernachlässigten Koeffizienten nach oben beschränkt ist:

$$\begin{aligned} E_T(N) &\equiv |u(x) - u_N(x)| = \left| \sum_{n=0}^{\infty} a_n T_n(x) - \sum_{n=0}^N a_n T_n(x) \right| = \left| \sum_{n=N+1}^{\infty} a_n T_n(x) \right| \\ &\leq \sum_{n=N+1}^{\infty} |a_n T_n(x)| \stackrel{|T_n(x)| \leq 1}{\leq} \sum_{n=N+1}^{\infty} |a_n|. \end{aligned} \quad (2.37)$$

Auch diese Formel ist für unbekanntene Lösungen nicht anwendbar, sondern wird durch eine zweite Faustregel ersetzt. Sie besagt, dass der numerische Trunkierungsfehler bei geometrisch konvergierenden Reihen etwa die Größenordnung des letzten Koeffizienten besitzt. Um dies zu zeigen wird Gleichung (2.35) in Gleichung (2.37) eingesetzt und ausgenutzt, dass der exponentielle Index der Konvergenz bei geometrischen Reihen $r=1$ beträgt:

$$E_T(N) \approx \sum_{n=N+1}^{\infty} |a_n| = \sum_{n=N+1}^{\infty} e^{-qn} = e^{-q(N+1)} \sum_{n=0}^{\infty} (e^{-q})^n = e^{-qN} \underbrace{\frac{e^{-q}}{1 - e^{-q}}}_{= \mathcal{O}(1)} = \mathcal{O}(e^{-qN}) = \mathcal{O}(|a_N|). \quad (2.38)$$

Unter Verwendung normierter Basispolynome kann mit diesem Fehler direkt derjenige des Funktionswertes abgeschätzt werden. Wenn a_N nur 1-2 Größenordnungen kleiner als a_0 ist, darf man von dem Ergebnis nicht mehr als eine exakte Dezimalziffer erwarten. Ist hingegen $a_N \approx \mathcal{O}(10^{-8})a_0$, so ist mit vielen korrekten Ziffern zu rechnen. Letztlich kann jedoch keine Fehlerabschätzung den universellen Test, die Berechnung mit anderem N zu wiederholen und die Ergebnisse zu vergleichen, ersetzen.

Manchmal ist es am effizientesten nur die räumliche Richtung spektral zu behandeln und entlang der Zeit t bzw. Skala k schrittweise zu integrieren. Mit dieser Methode zerfällt das partielle Differenzialgleichungssystem in ein System gewöhnlicher Differenzialgleichungen. In Kapitel 4 zeigt der Vergleich zwischen konventionellen und spektralen Integratoren, welche Variante für die Flussgleichung des anharmonischen Oszillator Vorteile besitzt.

Ein weiterer Vorteil der Spektralmethoden besteht darin, dass sie wegen ihrer Genauigkeit speichersparsam sind. Denn der Speicher ist eine deutlich stärkere Beschränkung als die Rechenzeit. Während es einfach ist, eine Berechnung etwas länger laufen zu lassen, führt fehlender Arbeitsspeicher zu einem gravierenden Problem. In diesem Fall muss unweigerlich zu einem anderen Algorithmus, wie einer Spektralmethode, gewechselt werden.

^[2] Die numerische Trunkierung darf nicht mit der physikalischen Trunkierung, d. h. der Projektion auf einen endlichdimensionalen Unterraum, oder der Trunkierung des Simulationsgebietes durch q_{\max} verwechselt werden.

3. Implementierung

Im ersten Abschnitt 3.1 dieses Kapitels werden die zur Implementierung notwendigen Formeln hergeleitet. Dazu gehört auch die Vorstellung notwendiger Matrizen, genutzter Kollokationspunkte und deren zugrunde liegenden Substitutionen. Abschließend wird das NEWTON-RAPHSON-Verfahren für die spektrale k -Integration erläutert. Der nachfolgende Abschnitt 3.2 enthält ergänzende Erklärungen zu der Umsetzung in MATLAB. Appendix A beinhaltet darüber hinaus einige Quelltextausschnitte sowie eine Übersicht aller m-Dateien.

3.1. Herleitungen

Zur Lösung der Flussgleichung mittels Kollokationsmethode dient der Ansatz

$$u_k(x) = \sum_{n=0}^{N_q} a_n(k) B_n(x) \quad (3.1)$$

mit den orthogonalen Basispolynomen $B_n \in \{T_n, P_n, H_n, L_n\}$. Namentlich sind dies die CHEBYSHEV-, LEGENDRE-, HERMITE- und LAGUERRE-Polynome. Die skalenabhängigen Koeffizienten $a_n(k)$ enthalten dabei die Fluktuationen. Durch die Wahl eines zum Ursprung symmetrischen Anfangspotenzials $V(q) = u_\Lambda(q)$ und die Form der Flussgleichung (2.30) bleibt das Potenzial $u_k(q)$ bei allen Skalen k gerade. Mit den Substitutionen

$$[0, q_{\max}^2] \ni q^2 = q_{\max}^2 \cdot \frac{\rho + 1}{2} \quad \Leftrightarrow \quad [-1, 1] \ni \rho = \left(\frac{q}{q_{\max}} \right)^2 \cdot 2 - 1 \quad \text{für } T_n, P_n \quad (3.2)$$

$$[0, q_{\max}^2] \ni q^2 = q_{\max}^2 \cdot \frac{\rho}{\rho_{\max}} \quad \Leftrightarrow \quad [0, \rho_{\max}] \ni \rho = \left(\frac{q}{q_{\max}} \right)^2 \cdot \rho_{\max} \quad \text{für } L_n \quad (3.3)$$

kann diese Symmetrie effektiv genutzt werden. Ohne diese wären alle ungerade Koeffizienten von $u_k(x)$ trivialerweise null und damit sowohl Speicherplatz als auch Rechenzeit verschwendet. Aufgrund der Substitutionen tragen in $u_k(\rho)$ auch die ungeraden Koeffizienten Informationen. Am Ende der Rechnung wird die andere Hälfte des Potenzials durch Spiegelung gewonnen. In der Konsequenz der Substitution lautet die Ortsdifferenziation

$$\frac{d}{dq} = \frac{4}{q_{\max}} \cdot \sqrt{\frac{\rho + 1}{2}} \frac{d}{d\rho}, \quad \frac{d^2}{dq^2} = \frac{4}{q_{\max}^2} \left(\frac{d}{d\rho} + 2 \cdot (\rho + 1) \frac{d^2}{d\rho^2} \right) \quad \text{für } T_n, P_n \quad (3.4)$$

$$\frac{d}{dq} = \frac{2}{q_{\max}} \cdot \sqrt{\rho \cdot \rho_{\max}} \frac{d}{d\rho}, \quad \frac{d^2}{dq^2} = 2 \cdot \frac{\rho_{\max}}{q_{\max}^2} \left(\frac{d}{d\rho} + 2 \cdot \rho \frac{d^2}{d\rho^2} \right) \quad \text{für } L_n. \quad (3.5)$$

Eine Ausnahme bildet die HERMITE-Basis. Wenn ihren Stützstellen ebenfalls die Substitution der Gleichung (3.2) zugrunde gelegt wird, ergibt sich wie in Abbildung 6 dargestellt eine sehr ungünstige Verteilung, der es an Gitterstellen in der Nähe des Ursprungs mangelt. Um die daraus resultierenden, schlechten Ergebnisse zu vermeiden, muss eine andere Substitution in Kauf genommen werden. Die Substitution

$$[-q_{\max}, q_{\max}] \ni q = q_{\max} \cdot \rho \quad \Leftrightarrow \quad [-1, 1] \ni \rho = \frac{q}{q_{\max}} \quad \text{für } H_n \quad (3.6)$$

berücksichtigt die Symmetrie nicht explizit, wodurch die Hälfte der Koeffizienten keine Information mehr tragen.

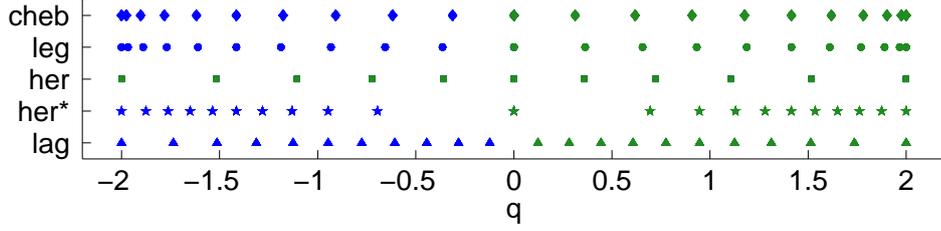


Abb. 6: Verteilung der Stützstellen in den verschiedenen Basen mit den Parametern $N_q = 10$ & $q_{max} = 2$. In Grün sind die „echten“ Kollokationspunkte dargestellt, mit welchen gerechnet wird. Ihre gespiegelten Partner sind in Blau abgebildet. Für die HERMITE-Basis sind sowohl die ungünstige Wahl her^* basierend auf der Substitution (3.2) als auch die verwendete Variante her nach Gleichung (3.6) gezeigt.

Wie bereits in Kapitel 2.3 erwähnt, existieren für jede Spektralbasis ausgezeichnete Kollokationspunkte, um die spektrale Konvergenz zu gewährleisten. Für die CHEBYSHEV-Polynome T_n wird das CHEBYSHEV-LOBATTO-Gitter

$$\rho_i = \cos\left(\frac{\pi i}{N_q}\right), \quad i = 0, \dots, N_q \quad (3.7)$$

gewählt. Im Fall der LEGENDRE-Polynome P_n dienen LEGENDRE-LOBATTO-Punkte

$$\rho_0 = 1, \quad \rho_{N_q} = -1, \quad \{\rho_i\} = \left\{ \rho : \frac{dP_{N_q}(\rho)}{d\rho} = 0 \right\}, \quad i = 1, \dots, N_q - 1 \quad (3.8)$$

als Stützstellen, wobei diese, wie bei allen Basen, absteigend sortiert sind. Sie liegen im Intervall $[-1, 1]$ und sind nicht in geschlossener Form für ein allgemeines N_q bekannt. Weil die *double precision* von MATLAB für große N_q bei der Nullstellenberechnung an ihre Grenze stößt, wurden die Nullstellen mit Mathematica und mindestens 40 Ziffern Genauigkeit vorab berechnet. Bei den HERMITE-Polynome H_n sind die Kollokationspunkte

$$\{\rho_i\} = \left\{ \frac{\rho}{\max_i |\rho_i|} : H_{N_q+1}(\rho) = 0 \right\}, \quad i = 0, \dots, N_q \quad (3.9)$$

auf das Intervall $[-1, 1]$ normiert und müssen ebenfalls eingelesen werden. Demgegenüber sind die Gitterstellen der LAGUERRE-Polynome L_n

$$\{\rho_i\} = \{\rho : L_{N_q+1}(\rho) = 0\}, \quad i = 0, \dots, N_q \quad (3.10)$$

nicht normiert, damit die darauf beruhende Matrix \mathbf{M}_q (wird im Folgenden eingeführt) bei wachsendem N_q nicht zu früh numerisch singular wird. Zur Berechnung der Basispolynome dienen folgende explizite Darstellungen:

$$T_n(x) = \cos(n \arccos(x)) \quad , \text{ Gleichung (19.191) in [12],} \quad (3.11)$$

$$P_n(x) = \frac{1}{2^n} \sum_{m=0}^{\lfloor \frac{n}{2} \rfloor} \frac{(-1)^m (2n-2m)!}{(n-m)! (n-2m)! m!} x^{n-2m} \quad , \text{ Gleichung (31) in [13],} \quad (3.12)$$

$$H_n(x) = n! \sum_{m=0}^{\lfloor \frac{n}{2} \rfloor} \frac{(-1)^m}{m! (n-2m)!} (2x)^{n-2m} \quad , \text{ Gleichung (22.3.10) in [14],} \quad (3.13)$$

$$L_n(x) = \sum_{m=0}^n \binom{n}{m} \frac{(-1)^m}{m!} x^m \quad , \text{ Gleichung (22.3.9) in [14].} \quad (3.14)$$

Eine wichtige Eigenschaft der Spektralmethoden ist die Äquivalenz der Lösungsdarstellung durch die Koeffizienten a_n und die Funktionswerte $u_k(\rho_i)$ an den Stützstellen. Mit der Umrechnungsmatrix \mathbf{M}_q ^[3] kann zwischen den Koeffizienten $\vec{a}_k = (a_0(k), \dots, a_N(k))^\top$ und Funktionswerten $\vec{u}_k = (u_k(\rho_0), \dots, u_k(\rho_N))^\top$ umgerechnet werden. Für die CHEBYSHEV-Polynome ist zur einfachen Berechnung der Funktionswerte

$$T_n(\cos(t)) = \cos(nt) \quad (3.15)$$

hilfreich, womit sich die Umrechnung ergibt:

$$\vec{u}_k = \mathbf{M}_q^{\text{cheb}} \cdot \vec{a}_k \text{ mit den Matrixeinträgen } (\mathbf{M}_q^{\text{cheb}})_{in} = \cos\left(n \frac{\pi i}{N_q}\right), \quad i, n = 0, \dots, N. \quad (3.16)$$

Analog ergeben sich die Matrizen $(\mathbf{M}_q)_{in} = B_n(\rho_i)$ für die anderen Basen. Für die umgekehrte Umrechnung dient die inverse Matrix \mathbf{M}_q^{-1} , die insbesondere bei den konventionellen Integratoren häufig benötigt wird. Zwecks höherer Genauigkeit wurden die inversen Matrizen bereits im Voraus mit Mathematica berechnet und als *comma-separated values* (csv) exportiert. MATLAB bietet mit seiner *double precision* nicht die notwendigen Voraussetzungen, um bei großen N_q die Inverse korrekt zu berechnen. Die Kosten für die Umrechnung von \vec{a}_k zu \vec{u}_k und umgekehrt betragen $\mathcal{O}(N^2)$.

Zur Lösung der Flussgleichung (2.30) wird die Differenziation des Potenzials benötigt. Eine elegante Variante die Ableitung^[4] $u'_k(q) = \sum_{n=0}^{N_q-1} \tilde{a}_n(k) B_n(q)$ zu berechnen ist die Multiplikation des Koeffizientenvektors mit der Differenziationsmatrix der CHEBYSHEV-Basis:

$$\tilde{a}_k = \mathbf{D}_q \cdot \vec{a}_k \text{ mit } \mathbf{D}_q^{\text{cheb}} = \begin{bmatrix} 0 & 1 & 0 & 3 & 0 & 5 & 0 & 7 & \dots \\ & 0 & 4 & 0 & 8 & 0 & 12 & 0 & \dots \\ & & 0 & 6 & 0 & 10 & 0 & 14 & \dots \\ & & & 0 & 8 & 0 & 12 & 0 & \dots \\ & & & & 0 & 10 & 0 & 14 & \dots \\ & & & & & 0 & 12 & 0 & \dots \\ & & & & & & 0 & 14 & \dots \\ & & & & & & & 0 & \dots \end{bmatrix}. \quad (3.17)$$

Zur einfacheren Handhabung wurde die Matrix durch die redundante letzte Zeile quadratisch gemacht. Für die LEGENDRE-Basis, die HERMITE-Basis und die LAGUERRE-Basis lauten die Analoga

$$\mathbf{D}_q^{\text{leg}} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & \dots \\ & 0 & 3 & 0 & 3 & 0 & \dots \\ & & 0 & 5 & 0 & 5 & \dots \\ & & & 0 & 7 & 0 & \dots \\ & & & & 0 & 9 & \dots \\ & & & & & 0 & \dots \end{bmatrix}, \quad \mathbf{D}_q^{\text{her}} = \begin{bmatrix} 0 & 2 & 0 & \dots \\ & 0 & 4 & 0 & \dots \\ & & 0 & 6 & \dots \\ & & & \dots & \dots \\ & & & & 0 & \dots \end{bmatrix}, \quad \mathbf{D}_q^{\text{lag}} = \begin{bmatrix} 0 & -1 & -1 & -1 & \dots \\ & 0 & -1 & -1 & \dots \\ & & 0 & -1 & \dots \\ & & & \dots & \dots \\ & & & & 0 & \dots \end{bmatrix}. \quad (3.18)$$

In der Matrixdarstellung ergibt sich damit die zweite Differenziation an den Stützstellen unter Verwendung der Gleichung (3.4):

$$\frac{d^2 u_k}{dq^2} \Big|_{\rho=\rho_i} = \frac{4}{q_{\text{max}}^2} \left(\sum_{n=0}^{N_q} (\mathbf{D}_q \cdot \vec{a}_k)_n T_n(\rho_i) + 2 \cdot (\rho_i + 1) \sum_{n=0}^{N_q} (\mathbf{D}_q \cdot \mathbf{D}_q \cdot \vec{a}_k)_n T_n(\rho_i) \right). \quad (3.19)$$

^[3] Der Index q bzw. k an den Matrizen hilft bei der späteren Unterscheidung zwischen den orts- und skalenbezogenen Varianten.

^[4] Das Prime-Zeichen ' kennzeichnet die Ableitung nach dem Argument der Funktion. Im Gegensatz dazu bezeichnet der Punkt ' im Folgenden die Differenziation ∂_k .

Somit lässt sich mit der Diagonalmatrix

$$\mathbf{diag}(\{\rho_i\} + 1) = \begin{bmatrix} \rho_0 + 1 & & \\ & \ddots & \\ & & \rho_N + 1 \end{bmatrix} \quad (3.20)$$

die Flussgleichung (2.30) in kompakter Matrixform schreiben

$$\left(\mathbf{M}_q \cdot \dot{\vec{a}}_k \right)_n = -\frac{1}{\pi} \frac{\frac{4}{q_{\max}^2} \left(\mathbf{M}_q \cdot \mathbf{D}_q \cdot \vec{a}_k + 2 \cdot \mathbf{diag}(\{\rho_i\} + 1) \cdot \mathbf{M}_q \cdot \mathbf{D}_q \cdot \mathbf{D}_q \cdot \vec{a}_k \right)_n}{k^2 + \frac{4}{q_{\max}^2} \left(\mathbf{M}_q \cdot \mathbf{D}_q \cdot \vec{a}_k + 2 \cdot \mathbf{diag}(\{\rho_i\} + 1) \cdot \mathbf{M}_q \cdot \mathbf{D}_q \cdot \mathbf{D}_q \cdot \vec{a}_k \right)_n} \equiv R(\rho_n). \quad (3.21)$$

Demnach lautet die Änderung der Koeffizienten

$$\dot{\vec{a}}_k = \mathbf{M}_q^{-1} \cdot \vec{R}. \quad (3.22)$$

Die komponentenweise Division der Gleichung (3.21) wird in MATLAB mit dem ./ Operator realisiert. Obwohl die Lösung des linearen Gleichungssystems mit dem MATLAB-Operator \ im Allgemeinen effizienter und genauer erfolgt, kann die explizite Verwendung der (mit Mathematica vorab exakt berechneten) inversen Matrix *Minv* von Vorteil sein. Wie ein Testlauf mit den Parametern *lambda*=1, *Nq*=20, *int*=4, *basis*='cheb' & *Nk* = 400 000 zeigt, lässt sich damit die Rechenzeit von 67s auf 14s senken.

Als Anfangsbedingung dienen die Koeffizienten des klassischen Potentials $V(q) = u_\Lambda(q)$, die mithilfe der Matrix \mathbf{M}_q^{-1} aus dem Vektor der Funktionswerte \vec{u}_Λ an den Stützstellen bestimmt werden.

Dieser Absatz stellt die Grundlagen und Formeln des Spektralintegrators vor. Bei der Weiterentwicklung des *k*-Integrators werden sowohl die Skalenachse als auch die *q*-Richtung spektral betrachtet. Dazu wird das in [15, 16] sowie im Appendix C von [9] erläuterte NEWTON-RAPHSON-Verfahren genutzt. Anhand der gewöhnlichen NEWTON-Iteration ist das Prinzip leicht nachvollziehbar. Beginnend mit einer TAYLOR-Entwicklung und einer anschließenden Linearisierung ist die Gleichung $f(x)=0$ mit der geschätzten Lösung $x \approx x^0$ folgendermaßen berechenbar:

$$f(x) = f(x^i) + \partial_x f(x^i)[x - x^i] + \mathcal{O}([x - x^i]^2). \quad (3.23)$$

Wenn die Anfangsschätzung exakt genug ist, können Terme zweiter und höherer Ordnung vernachlässigt und die lineare Gleichung

$$x^{i+1} = x^i - \frac{f(x^i)}{\partial_x f(x^i)} \quad (3.24)$$

so lange iteriert werden, bis die gewünschte Genauigkeit erreicht ist. Für das hier vorliegende Problem wird dieses Konzept von einer auf viele Unbekannte verallgemeinert. Dafür lautet der Spektralansatz

$$u(\rho, \kappa) = \sum_{n=0}^{N_q} \sum_{m=0}^{N_k} a_{nm} T_n(\rho) T_m(\kappa) \quad (3.25)$$

und die *k*-Substitution

$$[-1, 1] \ni \tilde{\kappa} = \frac{2k - (k_{\max} + k_{\min})}{k_{\max} - k_{\min}}, \quad k \in [k_{\min}, k_{\max}] \quad \Rightarrow \quad \frac{d}{dk} = \frac{2}{k_{\max} - k_{\min}} \frac{d}{d\tilde{\kappa}}. \quad (3.26)$$

Um einerseits das ganze Integrationsgebiet $k \in [0, \Lambda]$ abzudecken und andererseits genügend Stützstellen zu besitzen ohne eine zu große JACOBI-Matrix \mathbf{J} zu erhalten, werden Patches eingeführt. Diese Streifen $[k_{\min}, k_{\max}] \times [-q_{\max}, q_{\max}]$ unterteilen das Integrationsgebiet und auf

jedem dieser Streifen mit $N_k \lesssim 10$ wird nacheinander das Anfangswertproblem unter Beachtung eines stetigen Verlaufs gelöst. Wichtig ist diese Herangehensweise insbesondere bei Pseudospektralmethoden, da die dort auftretende JACOBI-Matrix \mathbf{J} vollbesetzt ist und die (in jedem Iterationsschritt notwendige) Berechnung ihrer Inversen $\mathcal{O}(N^3)$ Operationen benötigt. Mit der Kompaktifizierung

$$[-1, 1] \ni \kappa = 1 - \frac{2}{k+1}, \quad k \in [0, \infty) \quad \Rightarrow \quad \frac{d}{dk} = \frac{(1-\kappa)^2}{2} \frac{d}{d\kappa} \quad (3.27)$$

ist in einigen Fällen sogar ein einziger Patch ausreichend, weil die Kollokationspunkte an den relevanten Skalen dicht und bei großen Skalen, an denen kein signifikanter Fluss stattfindet, extrem weit verteilt liegen. Weitere notwendige Größen für den spektralen Integrator sind die Matrizen

$$(\mathbf{M}_k)_{in} = \cos\left(n \frac{\pi i}{N_k}\right) \quad \text{und} \quad \mathbf{D}_k = \mathbf{D}_q^\top. \quad (3.28)$$

Es ist zu beachten, dass die Differenziationsmatrix \mathbf{D}_k im Gegensatz zu ihrem Pendant \mathbf{D}_q von rechts auf die Koeffizientenmatrix \mathbf{A} wirken muss. Damit ergibt sich für die CHEBYSHEV-Variante mit Patches das Gleichungssystem

$$\vec{F}(\vec{A}) = 0, \quad \text{mit} \quad (3.29)$$

$$\vec{A} = (a_{00}, a_{01}, \dots, a_{0N_k}, a_{10}, a_{11}, \dots, a_{1N_k}, \dots, a_{N_q 0}, a_{N_q 1}, \dots, a_{N_q N_k})^\top \quad (3.30)$$

$$\vec{F} = (F_{00}, F_{01}, \dots, F_{0N_k}, F_{10}, F_{11}, \dots, F_{1N_k}, \dots, F_{N_q 0}, F_{N_q 1}, \dots, F_{N_q N_k})^\top \quad (3.31)$$

$$F_{ij}(\vec{A}) = \frac{2}{k_{\max} - k_{\min}} (\mathbf{M}_q)_{i:} \cdot \mathbf{A} \cdot \mathbf{D}_k \cdot (\mathbf{M}_k)_{j:}^\top - R_{ij} \Big|_{k_j = \frac{\kappa_j \cdot (k_{\max} - k_{\min}) + k_{\max} + k_{\min}}{2}} \quad (3.32)$$

$$R_{ij} = -\frac{1}{\pi} \frac{\frac{4}{q_{\max}^2} \left((\mathbf{M}_q)_{i:} \cdot \mathbf{D}_q \cdot \mathbf{A} \cdot (\mathbf{M}_k)_{j:}^\top + 2 \cdot (\rho_i + 1) \cdot (\mathbf{M}_q)_{i:} \cdot \mathbf{D}_q \cdot \mathbf{D}_q \cdot \mathbf{A} \cdot (\mathbf{M}_k)_{j:}^\top \right)}{k_j^2 + \frac{4}{q_{\max}^2} \left((\mathbf{M}_q)_{i:} \cdot \mathbf{D}_q \cdot \mathbf{A} \cdot (\mathbf{M}_k)_{j:}^\top + 2 \cdot (\rho_i + 1) \cdot (\mathbf{M}_q)_{i:} \cdot \mathbf{D}_q \cdot \mathbf{D}_q \cdot \mathbf{A} \cdot (\mathbf{M}_k)_{j:}^\top \right)}. \quad (3.33)$$

Angelehnt an die MATLAB-Syntax bezeichnet $(\mathbf{M}_q)_{i:}$ die i -te Zeile der Matrix \mathbf{M}_q . Ausgehend von einer Anfangsschätzung \vec{A}^0 wird das Gleichungssystem (3.29) iterativ gelöst:

$$\vec{A}^{n+1} = \vec{A}^n + \Delta \vec{A}^n. \quad (3.34)$$

Dabei ist ersichtlich, dass die geschätzte Anfangsverteilung für das NEWTON-RAPHSON-Verfahren im $(N_q+1) \cdot (N_k+1)$ -dimensionalen Raum bereits in demjenigen Gebiet liegen muss, in dem die gesuchte Lösung attraktiv wirkt. Für die vorliegende Flussgleichung ist als Anfangsschätzung ausreichend, die Anfangsbedingung $u_\Lambda(q)$ für alle Skalen k anzunehmen. Ihre Änderung $\Delta \vec{A}^n$ ergibt sich durch die JACOBI-Matrix \mathbf{J} gemäß

$$\mathbf{J}(\vec{A}^n) \cdot \Delta \vec{A}^n = -\vec{F}(\vec{A}^n) \quad (3.35)$$

und konvergiert asymptotisch, d. h. in der Nähe der Lösung, quadratisch. Die Einträge der JACOBI-Matrix \mathbf{J} lauten

$$(\mathbf{J})_{nm} = \frac{\partial F_n}{\partial a_m}, \quad n, m = 0, \dots, (N_q+1) \cdot (N_k+1) - 1 \quad (3.36)$$

und können aufgrund der einfachen Form der Flussgleichung (2.30) analytisch berechnet werden. Zur schnellen Umrechnung zwischen den Multiindices aus Gleichung (3.30) & (3.31) und den Indices aus Gleichung (3.36) dient die Subroutine *index*. Umgekehrt wandelt die Routine *indices*

den einfachen Index in den entsprechenden Multiindex um ($n \mapsto n_1 n_2$). In der Linearkombination von $u(\rho, \kappa)$ (vgl. Gleichung (3.25)) verbleibt nach der $a_{m_1 m_2}$ -Differenziation nur der Term $T_{m_1}(\rho) T_{m_2}(\kappa)$. Unter Verwendung der Quotientenregel folgt

$$\frac{\partial F_{n_1 n_2}}{\partial a_{m_1 m_2}} = \frac{2}{k_{\max} - k_{\min}} (\mathbf{M}_q)_{n_1} \cdot \delta \mathbf{A}_{m_1 m_2} \cdot \mathbf{D}_k \cdot (\mathbf{M}_k)_{n_2}^\top + \frac{1}{\pi} \frac{4}{q_{\max}^2} k_{n_2}^2 \cdot \frac{(\mathbf{M}_q)_{n_1} \cdot \mathbf{D}_q \cdot \delta \mathbf{A}_{m_1 m_2} \cdot (\mathbf{M}_k)_{n_2}^\top + 2 \cdot (\rho_{n_1} + 1) \cdot (\mathbf{M}_q)_{n_1} \cdot \mathbf{D}_q \cdot \mathbf{D}_q \cdot \delta \mathbf{A}_{m_1 m_2} \cdot (\mathbf{M}_k)_{n_2}^\top}{\left[k_{n_2}^2 + \frac{4}{q_{\max}^2} \left((\mathbf{M}_q)_{n_1} \cdot \mathbf{D}_q \cdot \mathbf{A} \cdot (\mathbf{M}_k)_{n_2}^\top + 2 \cdot (\rho_{n_1} + 1) \cdot (\mathbf{M}_q)_{n_1} \cdot \mathbf{D}_q \cdot \mathbf{D}_q \cdot \mathbf{A} \cdot (\mathbf{M}_k)_{n_2}^\top \right) \right]^2} \quad (3.37)$$

mit der differenzierten Matrix

$$(\delta \mathbf{A}_{m_1 m_2})_{ij} = \begin{cases} 1 & \text{für } i = m_1 \wedge j = m_2 \\ 0 & \text{sonst.} \end{cases} \quad (3.38)$$

Damit die Anfangsbedingung $u_\Lambda(q)$ erfüllt ist, muss ihre Gültigkeit an den entsprechenden Stützstellen gefordert werden:

$$F_{i0} = u(\rho_i, \kappa_0) - u_\Lambda(\rho_i). \quad (3.39)$$

Natürlich betrifft dies in der Konsequenz auch die entsprechenden Einträge der JACOBI-Matrix. Am anderen Intervallende liegende Stützstellen, d. h. F_{iN_k} , werden wie alle anderen inneren Gitterpunkte behandelt.

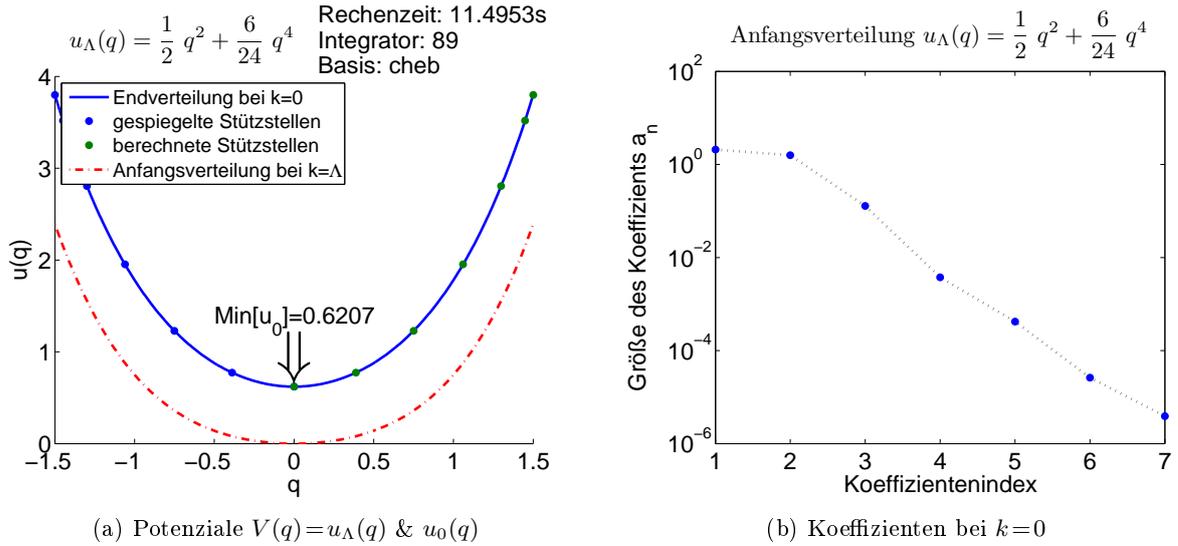
Ein Problem liegt in der oft schlechten Kondition großer Pseudospektralmatrizen, die mit wachsendem N_q zunimmt. In diesen Fällen ist die QR-Zerlegung, die $\mathbf{A} = \mathbf{Q} \cdot \mathbf{R}$ in das Produkt einer orthogonalen Matrix \mathbf{Q} mit einer oberen Dreiecksmatrix \mathbf{R} zerlegt, der LU-Zerlegung oder dem GAUSSSchen Eliminationsverfahren vorzuziehen, da es geringere Rundungsfehler erzeugt. Die QR-Zerlegung nutzt den Vorteil, dass die Kondition aller orthogonaler Matrizen eins ist, wodurch sich der numerische Fehler bei der Multiplikation mit \mathbf{Q} nicht erhöht.

Für kompliziertere Probleme muss gegebenenfalls mehr Aufwand bei der Umsetzung betrieben werden. Mögliche Anpassungen des Algorithmus sind eine genauere Anfangsschätzung \vec{A}^0 , die mit einem konventionellen Integrator berechnet wird, sowie die Bestimmung der JACOBI-Matrix \mathbf{J} durch eine Finite-Differenzen-Methode und die Optimierung des Sicherheitsfaktors *safety*.

Wie in [15] gezeigt wurde, hat die simultane Lösung auf dem ganzen Gebiet einen essenziellen Vorteil gegenüber der schrittweisen Lösung entlang der k -Ebenen. Demnach ist die COURANT-FRIEDRICHS-LEWY (CFL) Bedingung nicht zu berücksichtigen und somit ist es sogar möglich die Stützstellen in Raumrichtung q dichter zu wählen als diejenigen der Skala k .

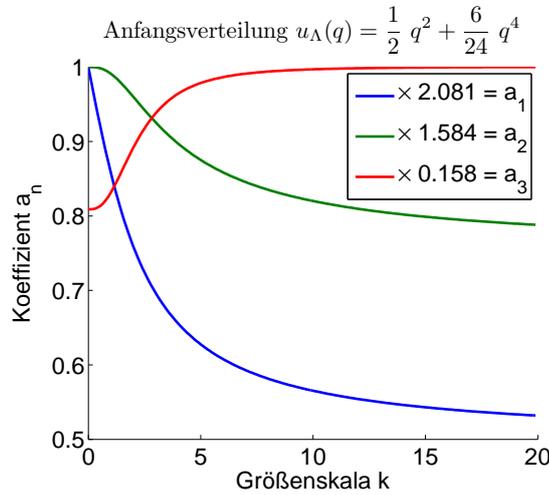
3.2. Umsetzung

Das Hauptprogramm *truncflowanho* (siehe Quelltextausschnitt 9 in Appendix A) erwartet vom Benutzer die im Folgenden genannten Parameter: Zur Festlegung des Anfangspotenzials $V(q) = \frac{1}{2} q^2 + \frac{\lambda}{24} q^4$ wird die Kopplung *lambda* eingelesen. Darüber hinaus bestimmt die Größe N_q den Grad des höchsten Polynoms und $N_q + 1$ die Anzahl der Basis-Polynome sowie der Stützstellen. Mit *qmax* wird die Gebietsgröße $[-q_{\max}, q_{\max}]$ festgelegt und die Variable *int* ist für die Wahl des Löser verantwortlich. Mögliche Optionen dafür sind *0* für das explizite EULER-Verfahren, *4* für das klassische explizite RUNGE-KUTTA-Verfahren 4. Ordnung, die zweistellige Zahl *xy* für den MATLAB-Löser *ode45* (die erste Ziffer legt die Option $RelTol = 10^{-x}$ und die zweite $AbsTol = 10^{-y}$ fest) und *100* für die spektrale Variante mit Patches sowie *200* für diejenige mit Kompaktifizierung. Durch *basis* wird die Wahl der Orthogonalbasis als String eingelesen, deren Möglichkeiten zwischen *cheb* CHEBYSHEV-, *leg* LEGENDRE-, *her* HERMITE- bzw. *lag* LAGUERRE-Polynome auswählt.



(a) Potenziale $V(q) = u_\Lambda(q)$ & $u_0(q)$

(b) Koeffizienten bei $k=0$



(c) Fluss der auf Eins normierten Koeffizienten innerhalb $k=40, \dots, 0$

Abb. 7: Ausgegebene Plots der Rechnung mit $\lambda = 6$, $Nq = 6$, $q_{\max} = 1,5$, $int = 89$ & $basis = 'cheb'$.

Ausgegeben werden vom Programm das Array u mit den Funktionswerten an den Kollokationspunkten bei der Skala $k=0$ und das Array a der dazugehörigen Koeffizienten. Außerdem gibt die Routine den Vektor $E = [E_0, E_1]$, bestehend aus der Energie des Grundzustandes und derjenigen des ersten angeregten Zustandes, zurück. Darüber hinaus gehören auch die Rechenzeit t zur Berechnung der k -Evolution und die Variable $gesiter$ mit der Anzahl der gesamten Iterationen zur Ergebnisausgabe. Exemplarisch zeigt Abbildung 7 die drei ausgegebenen Diagramme. Aus Plot (a) können alle Eingabeparameter abgelesen werden, ebenso wie die benötigte Rechenzeit t und die Grundzustandsenergie. In (b) ist der Abfall der Endkoeffizienten gezeigt. Da ihre Werte in der logarithmischen Darstellung in guter Näherung eine Gerade bilden, fallen die Koeffizienten exponentiell ab. Der Fluss der Koeffizienten in Abhängigkeit von k ist in Diagramm (c) dargestellt. Für $k \gg \omega$ ändern sich die Werte der Koeffizienten kaum, sondern erst wenn k die charakteristische Skala ω erreicht.

Im Folgenden wird auf die wichtigsten Zeilen des Hauptprogrammes *truncflowanho*, das in Appendix A Quelltextausschnitt 9 vorzufinden ist, kommentierend eingegangen. Zu Beginn werden die Variablen deklariert, wobei Zeile 7 von besonderer Bedeutung ist. Um ein w-förmiges Po-

tenzial zu erhalten, muss $\omega_2 = '1'$ gesetzt werden. Anschließend ruft das Programm in Zeile 11 die Subroutine *create_basis* auf, um entsprechend der Basiswahl *basis* die Stützstellen sowie die Matrizen Mq , $Minv$, Dq & Z zu erzeugen. In Zeile 25 beginnend findet die Berechnung der Anfangskoeffizienten statt. Danach startet die k -Evolution, deren Rechendauer mit der MATLAB-Funktion *tic...toc* gemessen wird. Für den *ode45*-Löser definiert Zeile 39 eine sogenannte anonyme Funktion, um der vom Integrator aufgerufenen Subroutine *RHS* die notwendigen Matrizen Z und $Minv$ als Parameter zu übergeben. Zeile 40 legt die Optionen des *ode45*-Lösers fest. Beruhend auf der Integratorwahl *int* legt dessen erste Ziffer die relative Toleranz und dessen zweite Ziffer die absolute Toleranz des Integrators fest. Außerdem wird die Statistik eingeschaltet, wodurch MATLAB die Anzahl der erfolgreichen und fehlgeschlagenen Schritte sowie die Zahl der Funktionsauswertungen auf das *Command Window* ausgibt. Mit diesen Optionen wird in Zeile 41 schließlich der Integrator ausgeführt. Dabei berechnet der RUNGE-KUTTA-Integrator vierter Ordnung mit Hilfe der fünften Ordnung eine Fehlerabschätzung und wählt die Schrittweite so, dass die Toleranz des Anwenders eingehalten wird. Wie exemplarisch in den Zeilen 45f. zu sehen, sollte die Initialisierung von Arrays stets explizit und in ihren endgültigen Dimensionen erfolgen, da ein nachträgliches Erweitern zwar möglich, jedoch langsam ist. Im Falle des spektralen Integrators erfolgt der Funktionsaufruf in Zeile 50 bzw. 57. In der Variante mit Patches erfolgt der Aufruf innerhalb der *for*-Schleife für jeden Streifen erneut. Ab Zeile 64 findet die basisabhängige Interpolation der Endkoeffizienten $a_n(k=0)$ statt, wobei die Resubstitution Beachtung findet. Zur Energieberechnung ab Zeile 103 muss in Zeile 73 (bzw. den analogen Zeilen der anderen Basen) die zweite Ableitung im Ursprung berechnet werden. Im Fall der CHEBYSHEV- & LEGENDRE-Basis wird dies durch die letzte Zeile der Matrix Z , die der letzten Stützstelle bei $q=0$ entspricht, und den Endkoeffizienten berechnet. Bei den HERMITE- und LAGUERRE-Polynomen ist eine Anpassung aufgrund der anderen Substitutionen sowie dem Fehlen der Stützstelle bei $q=0$ notwendig. Zum Schluss wird die Subroutine zum Plotten aufgerufen, auf die hier nicht näher eingegangen wird. Außerdem speichert das Programm mittels *save* alle eingeführten Variablen.

Quelltextausschnitt 1 des Appendixes A beinhaltet die Subroutine *create_basis*. Beruhend auf der Basiswahl *basis*, der Stützstellenanzahl Nq und der Gebietsgröße $qmax$ legt sie gemäß den Gleichungen aus Kapitel 3.1 die im Folgenden erläuterten Variablen an. Abschließend werden alle Größen an das Hauptprogramm *truncflowanho* zurückgegeben. Neben den Stützstellen ρ und q , die basierend auf den Substitutionen äquivalent zueinander sind, wird auch $qmax$ bestimmt. Für einen kompakten Quelltext ist es zweckmäßig, diese Variable unabhängig von der Basiswahl anzulegen, obwohl ihr Wert außer bei der LAGUERRE-Basis trivial 1 beträgt und nur bei der LAGUERRE-Substitution benötigt wird. Des Weiteren berechnet die Funktion die Matrizen Mq und deren Inverse $Minv$. Letztgenannte liest das Programm, außer bei den CHEBYSHEV-Polynomen, aus den mit Mathematica erzeugten *csv*-Dateien ein. Darüber hinaus werden die Differenzationsmatrix Dq und die Zählermatrix Z , die der zweiten Ableitung unter Berücksichtigung der Substitution entspricht, angelegt. Bei den Matrixmultiplikationen ist immer darauf zu achten, dass die Diagonalmatrix $\mathbf{diag}(\rho + 1)$ links von der Matrix Mq und diese wiederum links von der Differenzationsmatrix Dq steht.

Der Quelltextausschnitt 8 aus Appendix A enthält die Subroutine *spectralintegration* zur spektralen Integration ausgehend von den Anfangskoeffizienten $aStart$. Je nach ausgewählter Variante werden die k -Werte entweder in den Zeilen 3f. gemäß Gleichung (3.27) oder in den Zeilen 6ff. wie in Gleichung (3.26) angelegt. Danach belegt die *if*-Schleife die Faktoren der zweiten Ableitung basierend auf der Basiswahl (und der damit verbundenen Substitution) so, dass im Folgenden die Differenzation allgemeingültig geschrieben werden kann. Anschließend berechnet das Programm die Matrix Dk zur Differenzation in k -Richtung. Sie entspricht der transponierten Matrix Dq und muss von rechts auf die Koeffizientenmatrix A angewendet werden. Zeile 30 berechnet die Funktionswerte am Anfang des Patches beruhend auf den dortigen Koeffizienten $aStart$. Mit Zeile 32

wird ebenfalls auf diesen Koeffizienten basierend die Anfangsschätzung bestimmt. Innerhalb jeder Iteration des NEWTON-RAPHSON Verfahrens wird in den Zeilen 51ff. die JACOBI-Matrix J sowie in den Zeilen 86ff. der Vektor F ermittelt. Beiden Größen liegt die jeweils aktuelle Koeffizientenmatrix A zugrunde und in beiden Fällen müssen, wie in Kapitel 3.1 erläutert, die Anfangswerte getrennt betrachtet werden. Generell und insbesondere in Verbindung mit den hier aufgerufenen Subroutinen *index* (Quelltextausschnitt 3 in Appendix A) und *indices* (Quelltextausschnitt 4 in Appendix A) ist zu beachten, dass die Indizierung in MATLAB entgegen den Formeln dieser Arbeit mit Eins beginnt. Basierend auf den Indices der Basispolynome B_i , die bei Null beginnt, wurde für die Formeln jedoch die Konvention gewählt, dass die zugehörige Koeffizienten a_i den gleichen Index tragen.

Beginnend in Zeile 72 löst der Algorithmus das lineare Gleichungssystem. Wenn die in Zeile 74 ermittelte Kondition kleiner als 10^{10} ist, dient in Zeile 76 der \backslash Operator zur Auflösung nach der Änderung *delta*. Im Fall etwas schlechterer Kondition nutzt das Programm in Zeile 81 zur Verbesserung der Kondition eine Skalierung aller Zeilen auf Eins. Ist die Matrixkondition schlechter als 10^{14} , kommt in Zeile 78 die QR-Zerlegung zum Einsatz. Der Lösungsvektor *delta* wird im Anschluss durch die MATLAB-Routine *reshape* in Zeile 84 in seine Matrixform *Delta* transformiert. Daraus ergibt sich schließlich in Zeile 85, unter Berücksichtigung des Sicherheitsfaktors *safety*, die neue Koeffizientenmatrix A . Diese Iteration wird solange fortgeführt, bis entweder die in Zeile 96 ermittelte skalierte EUKLIDISCHE Norm

$$\|\vec{F}\| = \sqrt{\frac{\sum |F_i|^2}{(N_q + 1)(N_k + 1)}} \quad (3.40)$$

unter die Schwelle *tol* fällt oder die maximale Anzahl 100 erlaubter Iterationen erreicht wird. Die in Gleichung (3.40) enthaltene Skalierung ist notwendig, damit die Norm nicht von der Anzahl verwendeter Stützstellen abhängt.

4. Vergleich der Methoden

Dieser Abschnitt beinhaltet die Abhängigkeiten der Ergebnisse von den Eingabeparametern. Zu dieser Untersuchung werden sowohl der Integrator als auch die Gebietsgröße q_{max} , der Polynomgrad Nq und die Kopplungskonstante $lambda$ variiert. Außerdem kommen verschiedene Basispolynome $B_n \in \{T_n, P_n, H_n, L_n\}$ zum Einsatz. Der standardmäßige Werte der Kopplung $\omega^2 = 1$ wurde darüber hinaus in dem entsprechenden Absatz zur Analyse des w-förmigen Potentials negativ gewählt. Die unterschiedlichen Ausgabewerte werden jeweils vergleichend gegenübergestellt. In den unten stehenden Tabellen wird die auf unterschiedlichen Wegen berechnete Grundzustandsenergie und erste Anregungsenergie verglichen. Zum einen sind die Energien durch die hier vorgestellten Programme unter Verwendung verschiedener Integratoren berechnet. Zum anderen stammt ein Teil aus dem Polynomansatz $u_k(x) = \sum_{n=0}^N \frac{1}{(2n)!} a_{2n} x^{2n}$, der in die Flussgleichung (2.30) eingesetzt zu einem unendlichen System gekoppelter gewöhnlicher Differenzialgleichungen führt. In [1] wurde durch die Projektion auf Polynome vierten bzw. zwölften Grades das System trunziert und damit die Energien bestimmt. Die numerisch exakten Energiewerte entstanden aus der Diagonalisierung des Matrix-SCHRÖDINGER-Operators unter Verwendung der SLAC-Ableitung [17, 18]. Anstatt diskreter Differenzenoperatoren wird dabei die diskrete FOURIER-Transformation genutzt. Zur Berechnung der exakten Energien wurde der Quelltext aus [19] adaptiert.

Variation des Integrators

Bei der Untersuchung der Integratoren kamen unterschiedliche Vor- und Nachteile zum Vorschein. Tabelle 1 stellt die Resultate dieser Versuchsreihe gegenüber. Das explizite EULER-Verfahren ist ein Verfahren zweiter Ordnung und liefert sowohl für den harmonischen als auch für den anharmonischen Oszillator die schlechtesten Werte. Zwar lässt sich der relative Fehler durch Erhöhung der Stützstellenanzahl von 400 000 auf 2 000 000 um etwa den Faktor vier verkleinern, jedoch erhöht dies auch die Rechenzeit um etwa den Faktor fünf. Bessere Resultate erzeugt das klassische RUNGE-KUTTA-Verfahren vierter Ordnung, obwohl seine Rechenzeit nur etwa 50 % länger ist. Ihm reichen bereits 400 000 Stützstellen aus, um zuverlässige Resultate zu berechnen. Die Erhöhung ihrer Anzahl bewirkt keine deutliche Verbesserung. Bester konventioneller Integrator ist der in MATLAB integrierte *ode45*-Löser. Seine Rechenzeiten unter 20 s sind überzeugend und die Ergebnisse weichen beim harmonischen Oszillator maximal um 0,02 % bzw. beim anharmonischen Oszillator um höchstens 0,09 % von den exakten Werten ab.

Bei den spektralen Integratoren fällt als erstes die sehr kurze Rechenzeit auf. In der Variante mit Patching treten bereits in der zweiten Dezimalziffer Abweichungen zu den exakten Energiewerten auf. Wie Abbildung 8 verdeutlicht, ist dessen Ursache möglicherweise in der ungleichmäßigen

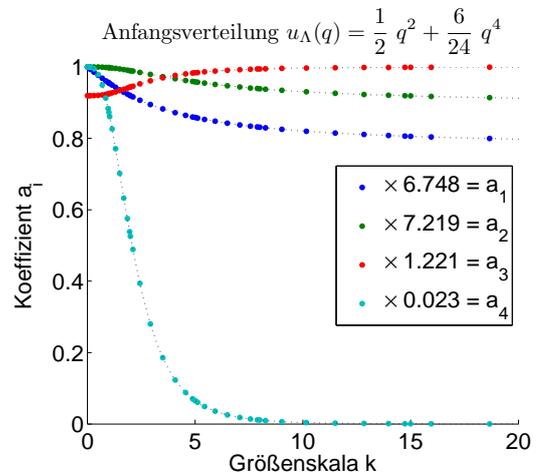


Abb. 8: Fluss der normierten Koeffizienten bei der Lösung mit dem spektralen Integrator 100 unter Verwendung von Patches. Die weiteren Eingabeparameter sind $lambda = 6$, $Nq = 10$, $Nk = 8$, $q_{max} = 2,5$ & $basis = 'cheb'$. An den Patchgrenzen $k = \{10000, 3000, 100, 40, 15, 8, 5, 2, 1, 0\}$ ist die Stützstellendichte deutlich erhöht.

Stützstellenverteilung verwurzelt, bei der sich die Stützstellen an den Grenzen der Patches konzentrieren. Demgegenüber berechnet die Version mit Kompaktifizierung überzeugende Resultate in der mit Abstand kürzesten Rechenzeit. Beruhend auf dieser Erkenntnis werden in den weiteren Vergleichen nur noch der *ode45*-Löser (mit den Standardparametern $RelTol = 10^{-8}$ & $AbsTol = 10^{-9}$) sowie der kompaktifizierte Spektralintegrator genutzt.

Tab. 1: Vergleich der verschiedenen Integratoren. In den ersten beiden Spalten sind der Integrator und die dazugehörige Stützstellenanzahl in k -Richtung genannt. Danach folgen in den Spalten drei bis fünf die Energien des Grundzustandes E_0 und des ersten angeregten Zustandes E_1 sowie die benötigte Rechenzeit T ^[5] für die Kopplungskonstante $\lambda = 0$. Dafür lauten die exakten Werte $E_0 = 0,5$ & $E_1 = 1,5$. Analog enthalten die Spalten sechs bis acht die entsprechenden Werte für $\lambda = 6$, wobei die exakten Energiewerte bei $E_0 = 0,6209$ & $E_1 = 2,0260$ liegen.

<i>int</i>	<i>Nk</i>	$\lambda = 0$			$\lambda = 6$		
		Energie E_0	Energie E_1	T [s]	Energie E_0	Energie E_1	T [s]
0	2000000	0,4992	1,4992	51,4	0,6199	2,0270	50,0
0	400000	0,4960	1,4960	10,6	0,6165	2,0239	10,6
4	2000000	0,5000	1,5000	75,5	0,6207	2,0278	73,5
4	400000	0,5000	1,5000	15,3	0,6207	2,0278	14,1
45	100000	0,5001	1,5001	18,4	0,6207	2,0278	20,2
89	100000	0,5000	1,5000	17,2	0,6207	2,0278	17,1
100	72	0,4995	1,4995	3,6	0,6197	2,0259	6,4
200	10	0,5000	1,5000	0,7	0,6207	2,0280	2,6

Variation der Gebietsgröße q_{\max}

Um einen möglichst geringen Fehler zu erzeugen, darf das Gebiet $[-q_{\max}, q_{\max}]$ erst bei ausreichend großem q_{\max} abgeschnitten werden. Im Gegenzug sind jedoch mit steigendem q_{\max} auch größere Nq notwendig, wie die folgende Untersuchung mit $Nq = 10$ verdeutlicht.

Für den harmonischen Oszillator ist die Lösung im Bereich $q_{\max} = 1, \dots, 40$ nicht von der Gebietsgröße abhängig. Bei größeren q_{\max} -Werten schwankt die Grundzustandsenergie um den exakten Wert. Dieser Effekt entsteht durch Rundungsfehler aufgrund der vielen Größenordnungen. Während die ersten beiden Koeffizienten die einzigen informationstragenden sind und in der Größenordnung $10^1 - 10^3$ liegen, beinhalten alle weiteren Koeffizienten nur numerische Nullen. Im Laufe der k -Evolution treffen die sehr unterschiedlichen Größenordnungen aufeinander und erzeugen dadurch Rundungsfehler. Der spektrale Integrator besitzt diese obere Schranke nicht, sondern löst alle Gebietsgrößen $q_{\max} > 1,5$ mit einer relativen Genauigkeit von etwa 0,003 %. Sein Vorteil liegt in der selbstkorrigierenden Funktionsweise. Alle Änderungen, die nicht zielführend waren, führen im nächsten Iterationsschritt des NEWTON-RAPHSON-Verfahrens zu einer größeren Abweichung, die wiederum automatisch minimiert wird.

Andere Beschränkungen zeigt das Potenzial des anharmonischen Oszillators. Hier werden sowohl mit dem *ode45*- als auch mit dem Spektralintegrator die Energien nur bei Gebietsgrößen zwischen 1,5-5 zufriedenstellend (mit einer maximalen relativen Abweichung 0,1 %) berechnet. Größere Gebiete oder anders ausgedrückt kleinere Stützstellendichten Nq/q_{\max} können den Fluss nicht ausreichend auflösen. Aus diesem Grund sollte dieser Quotient etwa 4-5 betragen. Durch die Nutzereingabe $q_{\max} = 0$ oder $Nq = 0$ wird dieser vorteilhafte Faktor vom Programm selbstständig gewählt. Wenn bei den folgenden Untersuchungen nichts anderes explizit genannt wird, so wurde von dieser Option Gebrauch gemacht.

^[5] Genutzter Prozessor: Intel Core i5 2430M mit 2 Kernen & 4 Threads bei 2,4-3,0 GHz.

Variation der Stützstellenanzahl N_q+1

Zuerst steht die Untersuchung des Stützstellenparameters beim *ode45*-Löser im Mittelpunkt. Die erste Messreihe mit $lambda=0$ offenbart einen verschwindenden Einfluss der Stützstellenanzahl auf das Resultat. Alle $N_q=2, \dots, 20$ liefern für die Grundzustandsenergie dasselbe Resultat mit einer relativen Abweichung 0,006 %. Ebenso variiert die Energie des ersten angeregten Zustandes nicht, für die der relative Fehler stets nur 0,002 % beträgt. Ab $N_q \geq 30$ wurde *RelTol* von 10^{-8} auf 10^{-6} und *AbsTol* von 10^{-9} auf 10^{-7} gesenkt, um die Rechenzeit in Grenzen zu halten. Denn die Laufzeit der Variante *int*=89 beträgt 79 min ohne ein besseres Resultat zu erzielen. Unabhängig von Toleranzwahl enthält die Lösung den in Abbildung 9(a) gezeigten problematischen Knick. Durch Akkumulation von Rundungsfehlern wachsen plötzlich die Koeffizienten a_3 , a_4 & a_5 an, obwohl sie das im Fall $lambda = 0$ nicht dürften. Entsprechend hat der zu Beginn genutzte Integrator *int*=89 Probleme diesen Knick einerseits sauber aufzulösen und andererseits mit seiner Genauigkeit zu vereinbaren, wodurch er dort hängen bleibt. Die *ode45*-Statistik mit $1,3 \cdot 10^7$ erfolgreichen gegenüber $7,3 \cdot 10^6$ missglückten Schritten bestätigt diese Probleme. Infolgedessen sind die Ergebnisse für $N_q \geq 30$ mit Vorsicht zu betrachten und vielmehr als Zufallstreffer zu bewerten. Abbildung 9(b) belegt die zunehmende Anzahl anwachsender Koeffizienten nochmals für das größere $N_q=80$. Im Spezialfall des harmonischen Potentials ist sogar $N_q=1$ ausreichend, um ohne Genauigkeitsverlust die Energiewerte zu berechnen, weil die LPA dafür keinen Fehler erzeugt und das quadratische Potenzial $u_k = a_0 + a_1 \rho = a_0 + a_1 \cdot [2(q/q_{\max})^2 - 1]$ bereits mit zwei Koeffizienten exakt darstellbar ist.

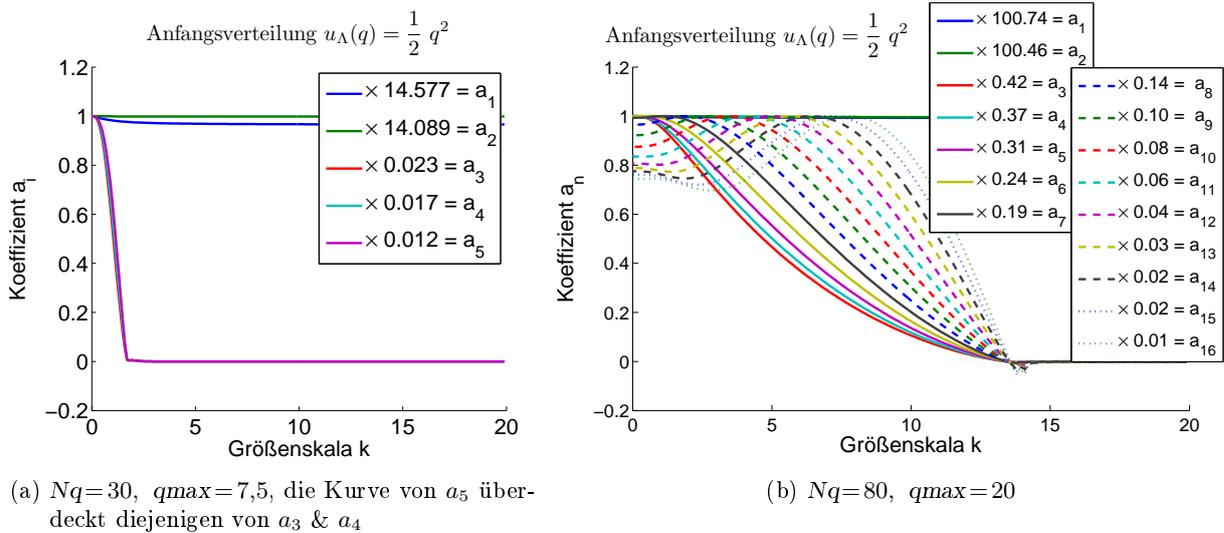


Abb. 9: Fluss der Koeffizienten für die Werte $lambda=0$, $int=67$.

Auch beim spektralen Integrator mit Kompaktifizierung zeigen sich dieselben Charakteristiken. Ab $N_q=2$ besitzen alle Ergebnisse eine gleichbleibende relative Abweichung von $\Delta E_0=0,003$ % bzw. $\Delta E_1=0,001$ %. Wie aus den Kurven der Laufzeiten in Abbildungen 10 zu entnehmen ist, steigt die Laufzeit ab $N_q=16$ deutlich an. Dessen Ursache besteht in den (asymptotischen) Kosten $\mathcal{O}(N^3)$ der LU- bzw. QR-Zerlegung, die mit wachsendem N_q rasch zunehmen. Zusätzlich verschlechtert sich die Kondition bei größer werdendem N_q , wodurch ansteigende Fehler und z. T. komplexe Energiewerte entstehen. Durch Oszillationen äußert sich diese Instabilität auch in den Koeffizientenverläufen.

Es stellt sich die Frage, ob auch für $lambda \neq 0$ bereits mit kleiner Stützstellenanzahl gute Werte erreichbar sind. Wie die Messreihe mit $lambda=6$ zeigt, liefern schon drei Kollokationspunkte ($N_q=2$) Ergebnisse mit maximaler relativer Abweichung von 0,1 %. Die Grundzustandsenergie

profitiert von einer weiteren Stützstelle (bzw. mit der damit verbundenen, größeren Gebietsgröße $q_{max} = \frac{Nq}{4}$) und besitzt für $Nq = 3, \dots, 80$ stets den maximalen relativen Fehler von 0,036 %. Demgegenüber bleibt die Energie des ersten angeregten Zustandes immer im Rahmen 0,091 %. Die hier ebenfalls bei großen Nq anwachsenden höheren Koeffizienten fallen nicht so stark ins Gewicht, da der relative Fehler von Anfang an um eine Größenordnung schlechter ist.

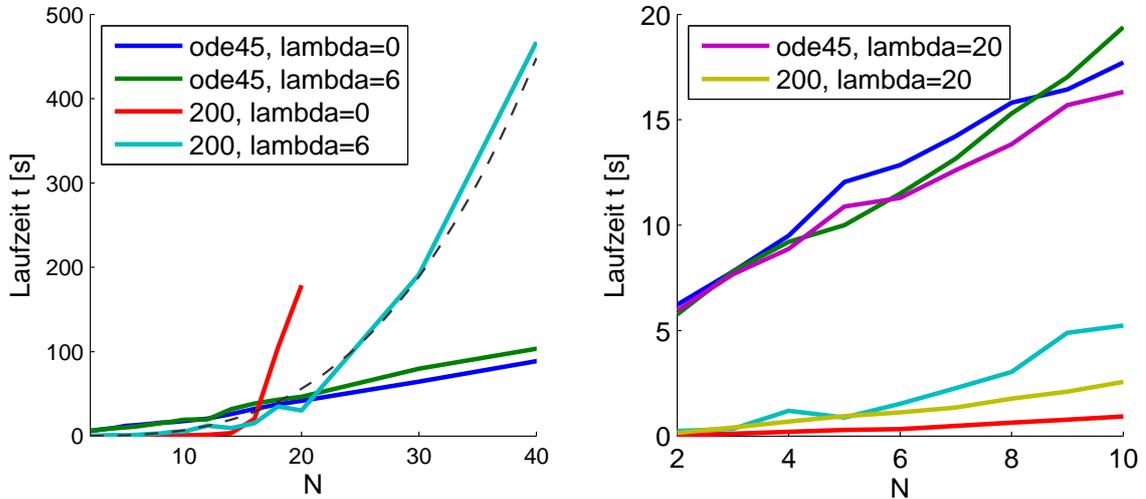


Abb. 10: Übersicht über die Laufzeiten der Messreihe unter Verwendung von CHEBYSHEV-Polynomen. Der Parameter 200 entspricht dem spektralen Integrator mit Kompaktifizierung. Im Gegensatz zum *ode45*-Löser zeigt der Spektralintegrator eine Abhängigkeit von λ . Im linken Plot wurde die rote Kurve abgebrochen, da für $Nq = 18$ komplexe Energiewerte entstehen. Die Übereinstimmung der Laufzeit der hellblauen Kurve mit der kubische Funktion $t(N) = 0,007 N^3$ bestätigt die erwarteten Kosten $\mathcal{O}(N^3)$. In der Vergrößerung rechts sind zusätzlich die zu $\lambda = 20$ gehörige Kurven eingezeichnet.

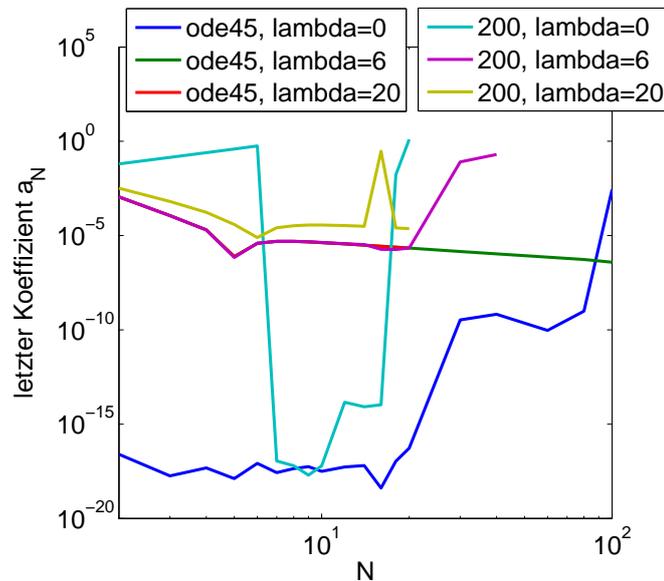


Abb. 11: In der Grafik ist der Betrag des letzten Koeffizienten über der Stützstellenanzahl aufgetragen. Bei den Berechnungen wurden die Parameter $int = \{89, 200\}$, $q_{max} = 0$ & $basis = 'cheb'$ genutzt. Die lila Kurve verdeckt z. T. die rote & grüne Kurve. Insbesondere bei den Kurven für $\lambda = 0$ wachsen die Koeffizienten mit steigendem Nq stark an. Die einzelnen Zacken sind numerische Artefakte.

Ähnliches ist beim spektralen Integrator zu beobachten. Dort liefert der Algorithmus bis $Nq \approx 20$ sehr genaue Ergebnisse. Danach wachsen ebenfalls die Fehler an. Wie bereits beim *ode45*-Löser beobachtet, profitiert die Genauigkeit lediglich von der dritten Stützstelle und alle weiteren bewirken keine signifikante Verbesserung, da im anharmonischen Fall die LPA die dominierende Fehlerursache darstellt. Beispielsweise durch eine Wellenfunktionsrenormierung $Z_k(q)$ könnte dieser Fehler reduziert werden.

Schließlich ist bei $lambda = 20$ das gleiche Schema zu erkennen. Optimal ist hier eine Stützstellenanzahl mit $Nq = 4 \dots 6$. Die relativen Fehler sind beim *ode45*-Löser $\Delta E_0 = 0,064\%$ & $\Delta E_1 = 0,172\%$ bzw. beim spektralen Integrator $\Delta E_0 = 0,044\%$ & $\Delta E_1 = 0,176\%$. Abbildung 11 zeigt den Verlauf der letzten Koeffizienten in Abhängigkeit der Stützstellenanzahl. In Übereinstimmung mit der Theorie gibt der letzte Koeffizient eine Abschätzung des Fehlers. Aus diesem Grund ist $lambda = 5$ ein guter Kompromiss zwischen Laufzeit und Präzision, weshalb dieser Wert für die Messreihe des folgenden Abschnitts gewählt wurde. Damit demonstriert der Spektralansatz seine Stärke, bereits bei kleinen Nq genaue Werte zu liefern. In Abbildung 10 sind die Rechenzeiten über der Stützstellenanzahl aufgetragen. Deutlich ist beim *ode45*-Löser der lineare Zusammenhang und die Unabhängigkeit von der Kopplungskonstanten $lambda$ erkennbar.

Die Engstellen des Verfahrens sind der Integrator, der die erreichbare Genauigkeit limitiert, sowie die Randeffekte der Gebietstrunkierung. Auch eine Erhöhung der Werte auf $RelTol = 10^{-12}$ & $AbsTol = 10^{-13}$ führen zu keiner Verbesserung. Im Fall $lambda \neq 0$ trägt auch die LPA entscheidend zum Gesamtfehler bei.

Variation der Kopplungskonstanten λ

Tab. 2: Übersichtstabelle der Messreihe zur Variation der Kopplungskonstanten $lambda$. Gegenübergestellt sind die Grundzustandsenergie und die Energie des ersten angeregten Zustandes. Die Vergleichswerte in der zweiten & siebten Spalte stammen aus [1] und wurden dort mit einem Polynomansatz vierten Grades *P4* gelöst. Selbiges trifft auf die Spalten drei & acht zu, wobei hierfür Polynome zwölften Grades *P12* angesetzt wurden. In den Spalten vier & neun sind die Ergebnisse des *ode45*-Lösers zu finden. Die Resultate des kompaktifizierten Spektralintegrators befinden sich in den Spalten fünf & zeh. Basierend auf der SLAC-Ableitung berechnete Werte sind in den Spalten sechs & elf gelistet.

λ	E_0					E_1				
	<i>P4</i>	<i>P12</i>	<i>ode45</i>	spek	exakt	<i>P4</i>	<i>P12</i>	<i>ode45</i>	spek	exakt
0	0,5000	0,5000	0,49997	0,50001	0,50000	1,5000	1,5000	1,49997	1,50001	1,50000
1	0,5277	0,5277	0,52768	0,52772	0,52774	1,6311	1,6315	1,63152	1,63156	1,63130
2	0,5506	0,5507	0,55070	0,55073	0,55079	1,7324	1,7341	1,73406	1,73411	1,73347
3	0,5706	0,5708	0,57082	0,57084	0,57095	1,8177	1,8207	1,82069	1,82075	1,81974
4	0,5885	0,5889	0,58889	0,58892	0,58906	1,8923	1,8968	1,89678	1,89687	1,89551
5	0,6049	0,6054	0,60541	0,60544	0,60561	1,9593	1,9652	1,96524	1,96535	1,96366
6	0,6201	0,6207	0,62070	0,62074	0,62093	2,0205	2,0278	2,02782	2,02796	2,02597
7	0,6343	0,6350	0,63498	0,63503	0,63524	2,0771	2,0857	2,08571	2,08587	2,08360
8	0,6476	0,6484	0,64842	0,64847	0,64870	2,1299	2,1397	2,13974	2,13992	2,13739
9	0,6602	0,6611	0,66114	0,66120	0,66144	2,1794	2,1905	2,19052	2,19071	2,18794
10	0,6721	0,6732	0,67322	0,67329	0,67355	2,2263	2,2385	2,23852	2,23873	2,23574
20	0,7694	0,7714	0,77142	0,77157	0,77191	2,5994	2,6209	2,62107	2,62119	2,61658

Im Vergleich der Werte aus Tabelle 2 demonstrieren die Spektralmethode mit *ode45*-Löser und der doppelt spektrale Algorithmus ihre Stärken. Beide Verfahren übertreffen nicht nur den Polynomansatz vierten Grades, sondern auch denjenigen zwölften Grades. Bei der Grundzustandsenergie liefern sie immer etwas exaktere Werte, wobei der Unterschied bei größeren $lambda$

zunimmt. Hingegen kann bei der ersten Anregungsenergie der Polynomansatz zwölften Grades in einigen wenigen Fällen die besten Resultate erzielen. Interessanterweise besitzt der vollspektrale Integrator exaktere Energiewerte für den Grundzustand und der *ode45*-Löser berechnet genauere Werte beim ersten Anregungszustand. Folglich kann das RUNGE-KUTTA-Verfahren die Krümmung des Potentials akkurater berechnen.

Zusammenfassend gesagt, erzeugen die drei Varianten *P12*, *ode45* und der spektrale Integrator gleich gute Ergebnisse, die sich meist in maximal einer Dezimalziffer unterscheiden. Alle drei Varianten können die exakten Werte, die unter Verwendung der SLAC-Ableitung aus den Eigenwerten der HAMILTONSchen Matrix gewonnen wurden, nicht erreichen. Dafür verantwortlich sind jedoch nicht die Integratoren, sondern die LPA, auf der alle drei Algorithmen beruhen.

Legendre-Basis $P_n(\rho)$

Zum Vergleich der verschiedenen Basispolynome wurden die Messreihen der Nq -Variation wiederholt. Die Integration für $lambda = \{0, 6, 20\}$ erfolgt sowohl mit dem *ode45*- als auch mit dem kompaktifizierten Spektralintegrator. Im Fall der LEGENDRE-Polynome besitzen alle Ergebnisse die gleichen relativen Abweichungen wie diejenigen der CHEBYSHEV-Basis.

Blendet man die LPA aus und betrachtet die Koeffizienten anstatt der Energiewerte, zeigt sich ein anderes Bild. In der Theorie besitzen die LEGENDRE-Polynome dieselbe Konvergenz, aber mit einem schlechteren algebraischen Vorfaktor. Dies drückt sich in einem um den Faktor \sqrt{N} schlechteren punktwisen Fehler aus. Abbildung 12 bestätigt diese Vorhersage in Form zweier Kurven. In der Konsequenz übersetzt sich die Differenz der Koeffizienten direkt in die Differenz der Fehler, weil die LEGENDRE-Polynome wie die CHEBYSHEV-Polynome durch die obere Schranke $|P_n(x)| \leq 1$ begrenzt sind.

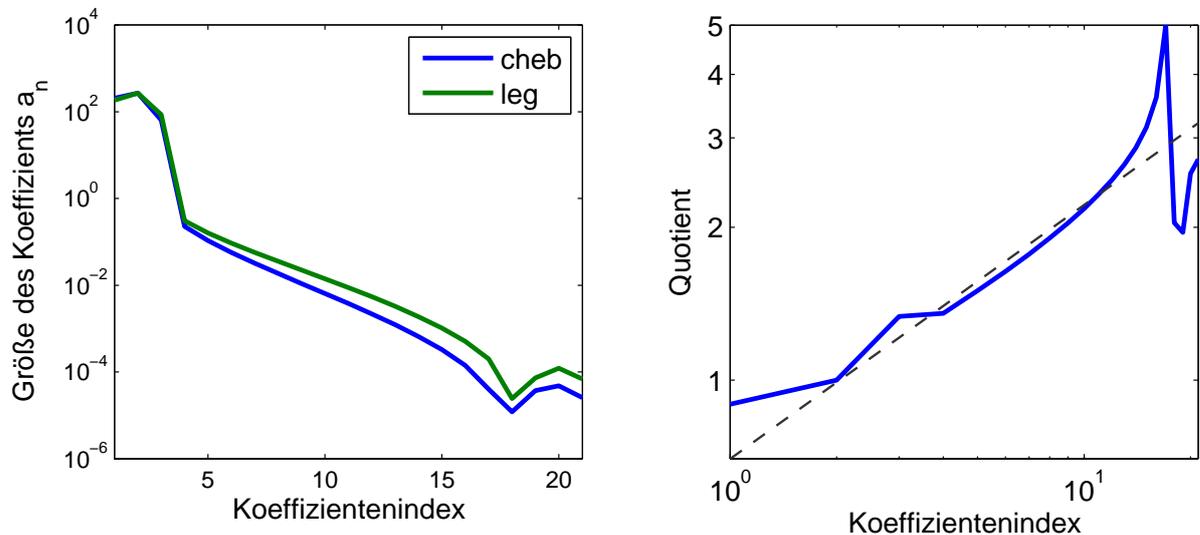


Abb. 12: Ergebnisse der Messung mit den Parametern $lambda = 20$, $Nq = 20$ & $qmax = 5$. Als Integrator wurde der *ode45*-Löser mit $RelTol = 10^{-12}$ & $AbsTol = 10^{-13}$ genutzt. Links sind die Absolutwerte der Koeffizienten bei $k=0$ halblogarithmisch aufgetragen. Der Knick beim Koeffizientenindex 18 ist ein numerisches Artefakt. Rechts ist doppelt logarithmisch die Wurzel ihres Quotienten dargestellt zusammen mit der gestrichelten Funktion $f(n) = 0,7\sqrt{n}$.

Diese Differenz kann auch auf eine andere Weise erklärt werden (vgl. Abbildung 13). Im Gegensatz zu den CHEBYSHEV-Polynomen, die über das ganze Intervall $[-1, 1]$ einheitlich oszillieren, besitzen die LEGENDRE-Polynome in den meisten Teilen des Intervalls eine kleine Amplitude, die nur nahe den Rändern anwächst. Jedoch dominiert dieser Effekt erst bei großen N_q . Deshalb ist seine Auswirkung auf die Spektralmethoden, die in der Regel stets bei $N_q \gtrsim 10$ praktiziert werden, gering.

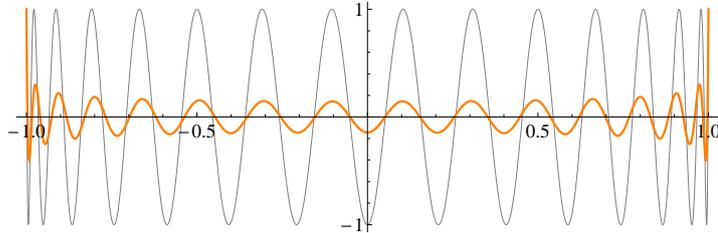


Abb. 13: Vergleich der Basispolynome $P_{30}(x)$ und $T_{30}(x)$. Während die Amplitude des LEGENDRE-Polynoms (orange) über das Intervall hinweg variiert, bleibt der Ausschlag des CHEBYSHEV-Polynoms (grau) konstant. Dies ist bereits aus der Eigenschaft $T_n(\cos(y)) = \cos(ny)$ direkt ersichtlich.

Hermite-Basis $H_n(\rho)$

Bei der Verwendung von HERMITE-Polynomen als Basis zeigen sich einige Unterschiede. Ab einer Stützstellenanzahl $\gtrsim 15$ wachsen die Fehler an. Dies äußert sich ab $N_q \gtrsim 19$ auch durch einen Knick im Fluss der Koeffizienten. In Abbildungen 14 ist die Oszillation der Endkoeffizienten auffallend. Ungeachtet dessen konvergieren die Koeffizienten spektral und das Auftreten dieses Effektes entsteht lediglich durch die anders gewählte Substitution bei der HERMITE-Basis. Dabei wird die Symmetrie des Potentials $u_k(q)$ nicht explizit genutzt, sondern äußert sich durch verschwindende ungerade Koeffizienten. Deren anfänglicher Wert beträgt exakt Null und wächst im Laufe der Integration numerisch leicht an.

Ebenfalls auf der anderen Substitution beruhend entsteht die Bedingung $N_q \gtrsim 4$. Beim *ode45*-Löser zeigt sich dies durch große Fehler unterhalb dieser Schranke und im Falle des kompaktifizierten Spektralintegrators bereits durch schlecht konditionierte JACOBI-Matrizen. Unter Berücksichtigung, dass der Koeffizient a_4 des q^4 bei der anderen Substitution dem Koeffizient a_2 des ρ^2 entspricht, ist diese Einschränkung nicht besonders gravierend. Anschaulich kommt dies dem Wegfallen der Stützstellen-Spiegelung gleich.

Darüber hinaus ist bei der HERMITE-Basis typisch, dass der letzte Koeffizient a_N mit steigendem Polynomgrad N_q nicht abfällt. Diese Tatsache wird durch das Anwachsen der anderen Koeffizienten a_n kompensiert, sodass sich die Anzahl der Größenordnungen zwischen a_0 & a_N effektiv verbessert.

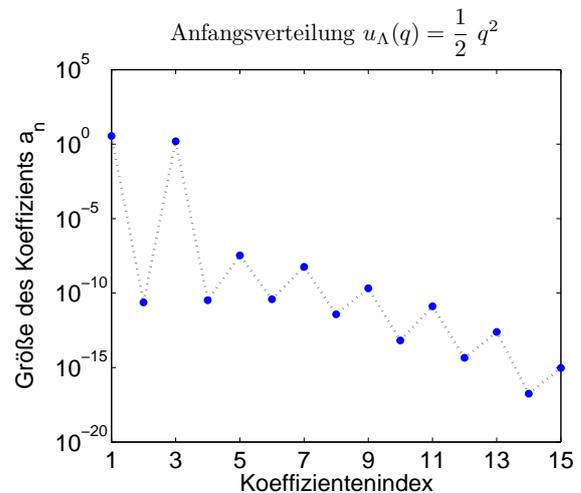


Abb. 14: Oszillierend abfallende Koeffizienten der HERMITE-Basis bei $k=0$. Die Parameter lauten $\lambda = 0$, $N_q = 14$, $q_{\max} = 3,5$ & $\text{int} = 89$.

Während der *ode45*-Löser für $Nq \gtrsim 4$ die gleiche Genauigkeit bei den Energiewerten erreicht wie die Referenz der CHEBYSHEV-Ergebnisse, kommen beim spektralen Integrator größere Einschränkungen zum Vorschein. Im Fall des harmonischen Potentials stimmen die Resultate für $Nq=4, \dots, 14$ überein, bei der Kopplungskonstanten $\lambda=6$ sind es noch die zu $Nq=6, \dots, 14$ gehörende Werte und bei $\lambda=20$ sogar nur diejenigen mit $Nq=8, \dots, 14$.

Laguerre-Basis $L_n(\rho)$

Wie bereits die HERMITE-Basis besitzt auch die LAGUERRE-Basis einige Einschränkungen gegenüber der CHEBYSHEV- & LEGENDRE-Basis. Beispielsweise entstehen bei großen Polynomgraden $Nq \gtrsim 24$ am Rand des Potentials Ausreißer. In Kombination mit dem *ode45*-Löser sind die Energiewerte bis zu dieser Schranke innerhalb einer Toleranz von 0,005 % in Übereinstimmung mit den Resultaten basierend auf den CHEBYSHEV-Polynomen. Für die Energie des ersten angeregten Zustandes sind die Werte sogar etwas exakter, weil die LAGUERRE-Basis in der Nähe des Ursprungs die meisten Stützstellen besitzt (vgl. Abbildung 6) und deshalb die Krümmung des Potentials am besten auflöst.

Zusammen mit dem spektralen Integrator und der Kompaktifizierung treten bereits früher Komplikationen auf. Obwohl dieses Verfahren die Energien des harmonischen Potentials bis $Nq \leq 16$ korrekt lösen kann, steigt bereits ab etwa zwölf Stützstellen die Rechenzeit deutlich – sowohl gegenüber kleineren Nq als auch im Vergleich mit den äquivalenten Rechnungen unter Verwendung von CHEBYSHEV-Polynomen. Noch früher verlieren die Ergebnisse des anharmonischen Potentials ihre Zuverlässigkeit. Ab etwa $Nq=12$ treten zum Teil komplexe Energiewerte, chaotische Potentiale $u_{k=0}(q)$, oszillierende Koeffizienten-Flüsse und überdurchschnittlich lange Rechenzeiten auf.

Negative Kopplung ω^2

Mit dem kompaktifizierten Spektralintegrator sind für $\omega^2 = -1$ keine Ergebnisse berechenbar, da seine JACOBI-Matrix \mathbf{J} stets schlecht konditioniert (Kondition $\gtrsim 10^{50}$) oder im Rahmen der Arbeitspräzision singular ist. Möglicherweise ist die Ursache in einer ungünstigen Anfangsschätzung für die NEWTON-RAPHSON-Iteration begründet. Durch die Annahme, dass anfänglich bei allen Skalen k das Potenzial $u_k(q)$ dem klassische Potenzial $V(q)$ entspricht, dessen zweite Ableitung im Ursprung negativ ist, kann der Nenner der Flussgleichung (2.30) Null werden und zu einer Divergenz führen. Durch eine vorkonditionierte Anfangsschätzung könnte diese Schwachstelle behoben werden. Bereits ohne Vorkonditionierung liefern die Variante mit Patches (9×10 Stützstellen entlang der Skalenachse) und der *ode45*-Löser brauchbare Resultate. Auch hier ist die Gebietsgröße $q_{max} = \frac{Nq}{4}$ eine gute Wahl und wird im Folgenden verwendet. Eine weitere Vorbetrachtung zeigt, dass für kleine Kopplungskonstanten λ mindestens neun Stützstellen notwendig sind, um sinnvolle Ergebnisse zu erreichen. Bei kleineren Potenzialbarrieren, d. h. größeren λ , würden vier Stützstellen ausreichen. Jedoch werden zur besseren Vergleichbarkeit der Messreihe durchgehend acht Kollokationspunkte genutzt. Tabelle 3 fasst die damit erhaltenen Energiewerte zusammen.

Zur Berechnung der Grundzustandsenergie eignen sich der Polynomansatz zwölften Grades sowie der spektral Ansatz in Kombination mit dem *ode45*-Löser. Bei Kopplungskonstanten $\lambda \leq 6$ ist ihre relative Abweichung vom exakten Wert meist etwa 2 % und bei größeren Kopplungskonstanten stets unter 0,75 %. Der spektrale Integrator mit Patches hat hingegen bei $\lambda < 4$ Probleme. Dies äußert sich in den komplexen Anteilen der Energiewerte (in Tabelle 3 sind nur die Realteile eingetragen), oszillierenden Potentials und Koeffizientenverläufen mit Knicken. Für flache Anfangspotentiale kann der spektrale Integrator jedoch zu seinen beiden zuvor genannten

Alternativen aufschließen und liefert dort sogar die genauesten Werte der ersten Anregungsenergie. Allerdings sind auch die Resultate des *ode45*-Integrators und des Polynomansatzes zwölften Grades nur geringfügig schlechter. Im Bereich kleiner Kopplungskonstanten kann der spektrale Ansatz in der *ode45*-Variante mit den besten Energiewerten des ersten angeregten Zustandes überzeugen. Auch bei der Energie der ersten Anregung nehmen die relativen Fehler mit wachsender Kopplungskonstante λ ab und der Polynomansatz vierten Grades besitzt die mit Abstand größten relative Abweichungen.

Tab. 3: Übersichtstabelle der Messreihe zur Variation der Kopplungskonstanten λ bei negativem $\omega^2 = -1$. Gegenübergestellt sind die Grundzustandsenergie und die Energie des ersten angeregten Zustandes. Die Werte des Polynomansatzes vierten Grades $P4$ und zwölften Grades sind aus [1] entnommen. Beruhend auf dem Spektralansatz achten Grades sind die Resultate sowohl durch den *ode45*-Löser als auch den Spektralintegrator in dessen Variante mit Patches berechnet. Erneut wurden mit dem Algorithmus basierend auf der SLAC-Ableitung die exakten Energiewerte ermittelt.

λ	E_0					E_1				
	$P4$	$P12$	<i>ode45</i>	spek	exakt	$P4$	$P12$	<i>ode45</i>	spek	exakt
1	-	-	-0,8797	-0,7596	-0,8556	-	-	-0,8043	-0,1281	-0,8299
2	-	-0,2474	-0,2481	-0,2136	-0,2422	-	0,0049	0,0061	-0,2136	-0,0216
3	0,2473	-0,0681	-0,0682	-0,0451	-0,0652	-0,2241	0,3514	0,3497	-0,0451	0,3307
4	-0,0186	0,0286	0,0287	0,0285	0,0308	0,3511	0,5753	0,5752	0,5746	0,5598
5	0,0654	0,0949	0,0949	0,0947	0,0967	0,5835	0,7455	0,7458	0,7451	0,7324
6	0,1234	0,1457	0,1457	0,1455	0,1472	0,7509	0,8842	0,8846	0,8838	0,8723
7	0,1688	0,1871	0,1871	0,1868	0,1885	0,8851	1,0021	1,0024	1,0016	0,9909
8	0,2063	0,2223	0,2223	0,2220	0,2236	0,9987	1,1052	1,1055	1,1046	1,0944
9	0,2671	0,2530	0,2530	0,2527	0,2543	1,1863	1,1972	1,1974	1,1964	1,1866
10	0,2386	0,2803	0,2803	0,2800	0,2816	1,0978	1,2805	1,2807	1,2796	1,2701
20	0,4536	0,4632	0,4632	0,4626	0,4643	1,7866	1,8638	1,8642	1,8625	1,8538

Die Trunkierung auf ein Potenzial vierten Grades liefert zwar für positives ω^2 ordentliche Werte, scheidet jedoch am w-förmigen Potenzial. Dafür ursächlich ist die für kleine λ/ω^3 starke Veränderung des effektiven Potenzials $u_k(q)$ nahe der Minima des klassischen Potenzials $V(q)$. Jedoch ist dies für die Polynom-Approximation, deren Gültigkeit im Ursprung liegt, nicht zu bewältigen. Während die Erhöhung des Polynomgrades von vier auf zwölf beim konvexen Potenzial $V(q)$ keine signifikante Verbesserung erzielte, profitiert das Ergebnis bei negativem ω^2 deutlich davon.

Auch die anderen Basispolynome liefern bei $\omega^2 = -1$ gute Werte. In Verbindung mit dem *ode45*-Löser sind die Resultate nicht von den Energiewerten der CHEBYSHEV-Basis zu unterscheiden. Mit leichten Schwankungen verbunden, stimmen die Resultate des spektralen Integrators aller vier Basen überein. Demnach existieren für $Nq=8$ keine ausgezeichneten Basispolynome.

Je größer die Barriere des klassischen Potenzials $V(q)$, d. h. je kleiner die Kopplung λ für festes $\omega^2 = -1$, desto schwieriger wird es, mit der LPA zuverlässige Energiewerte zu berechnen. Die Ursache dafür sind die nahezu entarteten Energieniveaus und die Dublettaufspaltung aufgrund von Instantonen-Effekte. Um diese Effekte aufzulösen, müsste über die führende LPA-Ordnung hinausgegangen werden [20].

Fazit der numerischen Resultate

Mit der MATLAB-Routine *truncflowanho* können neben den Energiewerten des Grundzustandes und des ersten angeregten Zustandes auch der Fluss des Potentials (siehe Abbildung 2) und der Koeffizienten (vgl. Abbildung 7 (c)) mit guter Präzision berechnet werden. Obwohl die Variante der Routine *Hmatrix* unter Verwendung der SLAC-Ableitung numerisch die exakten Werte in einer deutlich kürzeren Laufzeit ermittelt, führen die Resultate der Flussgleichung zusätzlich zu physikalischem Verständnis. Aus der Skalenabhängigkeit des Potentials $u_k(q)$ und der Flussgleichung (2.30) folgt anschaulich die Anhebung des Potentials oder die Glättung des w-förmigen Potentials bis es schließlich seine konvexe Form bei $k=0$ annimmt. Darüber hinaus lässt sich der Fluss der Koeffizienten in den Fluss der Kopplungskonstanten umwandeln, wodurch die skalenabhängigen Kopplungen studiert werden können.

Prinzipiell sind mit allen diskutierten Varianten sinnvolle Ergebnisse möglich. Dennoch heben sich die CHEBYSHEV-Basispolynome hervor, da sie am einfachsten zu berechnen sind und die zu ihnen gehörenden Matrizen eine gute Kondition besitzen. Bei den Integratoren scheiden das explizite EULER-Verfahren und das klassische RUNGE-KUTTA-Verfahren 4. Ordnung infolge ihrer unzureichenden Genauigkeit bzw. Rechendauer aus. Am zuverlässigsten ist die MATLAB-Routine *ode45* mit ihrer Schrittweitensteuerung, die mit keinem der Testprobleme Schwierigkeiten hatte. Demgegenüber hat der spektrale Integrator den großen Vorteil der sehr kurzen Laufzeit und der nicht zu berücksichtigenden CFL-Bedingung. In den meisten Fällen stellt die kompaktifizierte Variante das Optimum mit unübertroffener Geschwindigkeit und akkuraten Resultaten dar. In ihrer Stützstellenverteilung ist berücksichtigt, dass der Fluss erst bei kleinen Skalen signifikant wird. Um ihre Probleme beim w-förmigen Potential zu beheben, müsste eine vorkonditionierte Anfangsschätzung implementiert werden.

5. Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde erfolgreich eine Routine zur Lösung des quantenmechanischen anharmonischen Oszillators in MATLAB implementiert. Neben den variablen Eingabeparametern, wie der Stützstellenanzahl N_q , der Gebietsgröße q_{\max} und der Kopplungskonstanten λ , können vom Benutzer die Basispolynome sowie der Integrator ausgewählt werden. Zu den Möglichkeiten gehören auf der einen Seite die CHEBYSHEV-, LEGENDRE-, HERMITE- und LAGUERRE-Polynome. Auf der anderen Seite stehen der explizite EULER-Integrator, das explizite klassische RUNGE-KUTTA-Verfahren vierter Ordnung, die MATLAB-Routine *ode45* mit Schrittweitensteuerung sowie der spektrale Integrator, beruhend auf der NEWTON-RAPHSON-Iteration, zur Verfügung. Beim Spektralintegrator kann zusätzlich zwischen den Varianten mit Patching oder einer Kompaktifizierung ausgewählt werden.

In umfangreichen Tests zeigen sich sowohl Stärken als auch Schwächen der verschiedenen Methoden. Die größte Stärke der Spektralmethoden ist die geringe Anzahl notwendiger Stützstellen. Davon profitiert der spektrale Integrator doppelt, denn bei ihm wird neben dem Ort auch die Skala k spektral behandelt. Infolgedessen ergeben sich damit nicht nur sehr gute Resultate, sondern auch äußerst kurze Laufzeiten, welche die konventionellen Varianten meist um etwa den Faktor 20 schlagen.

Der Einfluss der Gebietsgröße q_{\max} ist zum einen aufgrund des entstehenden Fehlers durch Trunkierung des Gebietes und zum anderen durch den Zusammenhang mit der Stützstellendichte entscheidend. Sobald die Gebietsgröße $q_{\max} \gtrsim 1$ und der Quotient $N_q/q_{\max} \approx 4$ beträgt, können alle relevanten Informationen korrekt aufgelöst und genaue Ergebnisse berechnet werden.

Erwartungsgemäß zeigt die Spektralmethode, dass sie bereits mit einer kleinen Anzahl Stützstellen $N_q \approx 5$ hervorragende Werte liefert. Größere N_q erzielen keine Verbesserungen, weil die Fehler abseits der Spektralmethode liegen. Im Falle des anharmonischen Oszillators stellt die *local potential approximation* den dominierenden Fehler dar. Ab $N_q \gtrsim 20$ steigen meist die Abweichungen zu den exakten Werten. Der Genauigkeitsverlust lässt sich oft bereits im Fluss der Koeffizienten erkennen, wenn Knicke oder Oszillationen auftreten. Außerdem sind stark anwachsende Rechenzeiten ein Anzeichen für numerische Schwierigkeiten, wie z. B. schlecht konditionierte Matrizen oder Probleme der Schrittweitensteuerung.

Unabhängig von der Kopplungskonstanten λ überzeugen die ermittelten Energiewerte der spektralen Algorithmen. Ohne Schwierigkeiten können sie die Werte des Polynomansatzes zwölften Grades reproduzieren. Obwohl der kompaktifizierte Spektralintegrator beim w-förmigen Potenzial ($\omega^2 = -1$) aufgrund seiner schlecht konditionierten JACOBI-Matrix \mathbf{J} erfolglos ist, demonstrieren die anderen Spektralmethoden ihre Stärke. Im Gegensatz zur Kompaktifizierung liefert die Version mit Patches zufriedenstellende Ergebnisse. Mit dem *ode45*-Löser können die exaktesten Energiewerte erreicht werden, die insbesondere bei kleinen Kopplungskonstanten λ selbst die Genauigkeit des Polynomansatz zwölften Grades übertreffen.

Unter Verwendung der LEGENDRE-Polynome lassen sich genauso exakte Energiewerte berechnen. Erst der direkte Vergleich der Koeffizienten offenbart eine um den Faktor $\sqrt{N_q}$ schlechtere asymptotische Konvergenz. Größere Einschränkungen weisen die HERMITE-Polynome auf. Durch ihre andere Substitution, welche die Spiegelsymmetrie nicht explizit ausnutzt, benötigen sie mindestens vier Stützstellen. Außerdem werden ihre Resultate bereits ab $N_q \gtrsim 15$ ungenauer. In Verbindung mit dem spektralen Integrator nimmt in Abhängigkeit der Kopplungskonstanten λ zudem der Übereinstimmungsbereich mit der CHEBYSHEV-Lösung ab. Ebenso haben die LAGUERRE-Polynome Nachteile gegenüber der CHEBYSHEV-Referenz. Einerseits können in Kombination mit dem *ode45*-Lösers bis $N \lesssim 24$ vergleichbare Energiewerte erreicht werden. Andererseits treten beim spektralen Integrator bereits ab $N \approx 12$ Komplikationen auf, die sich teilweise durch komplexe Energiewerte, chaotische Potenziale, oszillierende Koeffizientenverläufe und überdurchschnittlich lange Laufzeiten äußern.

Obwohl der Quelltext bereits überzeugende Resultate liefert, bieten sich noch Stellen zur Verbesserung an. Als Erstes fällt das Versagen des kompaktifizierten Spektralintegrators beim w-förmigen Potenzial negativ auf. Zur Lösung des Problems ist eine bessere Anfangsschätzung unter Verwendung eines Vorkonditionierers notwendig. Abgesehen davon könnte folgendes Konzept zur Verbesserung der Genauigkeit von Nutzen sein. Durch Teilung des Gebietes an den Minima des klassischen Potenzials $V(q)$ entstehen drei Intervalle. Im mittig gelegenen, endlichen Intervall erfährt das effektive Potenzial $u_k(q)$ die größte Veränderung und wird dort nahe $k = 0$ sehr flach. In diesem Gebiet sollten CHEBYSHEV-Polynome $T_n(q)$ angesetzt werden. Für die einseitig unbegrenzten Intervalle, in denen das Potenzial unbegrenzt anwächst, bieten sich LAGUERRE-Polynome $L_n(q)$ oder rationale CHEBYSHEV-Polynome $TL_n(x)$ an.

Darüber hinaus sind die Einschränkungen der HERMITE-Polynome $H_n(q)$ sicherlich optimierbar. Bei den HERMITE-Polynome wachsen die Faktoren sehr schnell an, wie das Polynom zwanzigsten Grades zeigt: $H_{20}(x) = 1\,048\,576x^{20} + \dots + 670\,442\,572\,800$. Dadurch entstehen bei wachsendem N_q bereits früh numerische Instabilitäten durch *Overflows*. Abhilfe könnten hier die besser konditionierten orthonormierten HERMITE-Funktionen schaffen, die nach oben durch $|\Psi_n(x)| \leq 0,816$ beschränkt sind. Auch die LAGUERRE-Polynome haben das Problem der Unbeschränktheit und sollten in einer Weiterentwicklung durch die LAGUERRE-Funktionen $\Phi_n(x) \equiv \exp[-\frac{x}{2}]L_n(x)$ ersetzt werden.

Zuletzt ist die inverse Matrix \mathbf{M}_q^{-1} eine Fehlerquelle, deren Hybridlösung (dem Einlesen der mittels Mathematicas höherer Genauigkeit berechneten und exportierten Matrizen) einen provisorischen Eindruck vermittelt. BOYD nennt in [9] eine Alternative zur Berechnung der inversen Matrix. Demnach muss die Matrix nicht numerisch invertiert werden, sondern ist auch explizit berechenbar. Ihre Elemente lauten

$$(\mathbf{M}_q^{-1})_{ij} = \frac{B_i(k_j) w_j}{(B_i, B_i)}, \quad (5.1)$$

wobei im Nenner das innere Produkt der i -ten Basisfunktion mit sich selbst vorkommt und im Zähler das Gewicht w_j der GAUSSschen Quadratur.

Als abschließendes Fazit muss nochmals die Stärke der CHEBYSHEV-Polynome hervorgehoben werden. Keine andere Basis erreicht durchgängig die mit ihnen erzielte Genauigkeit. Wie das Eingangszitat von JOHN P. BOYD bereits zum Ausdruck bringt, ist diese Basiswahl für nichtperiodische Probleme in der Regel optimal.

A. Quelltextausschnitte

Im Folgenden werden die wichtigsten Quelltextausschnitte aufgelistet. Damit die Übersichtlichkeit gewährleistet bleibt, werden unwichtige Passagen mit [...] ausgespart und Quelltext-Kommentare ohne Kennzeichnung gekürzt oder ausgelassen. Weiterführende Bemerkungen zur Implementierung sind in Kapitel 3.2 zu finden. Auf der beiliegenden CD sind alle Quelltextdateien und benötigten Textdateien enthalten:

- *create_basis.m*: Berechnet Stützstellen und Matrizen \mathbf{M}_q , \mathbf{M}_q^{-1} , \mathbf{D}_q & \mathbf{Z} bzw. liest sie ein
- *ergplot.m*: Erzeugt Ergebnis-Plots und speichert diese auf Wunsch in den Ordner *Output*
- *Hermite.m*: Berechnet Koeffizienten des n -ten HERMITE-Polynoms $H_n(x) = c_n x^n + \dots + c_0$
- *index.m*: Berechnet aus den Matrix-Indices (n,m) den Index p des Vektors
- *indices.m*: Berechnet aus dem Vektor-Index p die Matrix-Indices (n,m)
- *Laguerre.m*: Berechnet Koeffizienten des n -ten LAGUERRE-Polynoms $L_n(x) = c_n x^n + \dots + c_0$
- *Legendre.m*: Berechnet Koeffizienten des n -ten LEGENDRE-Polynoms $P_n(x) = c_n x^n + \dots + c_0$
- *RHS.m*: Berechnet rechte Seite des Differenzialgleichungssystems für den *ode45*-Löser
- *spectralintegration.m*: Enthält den spektralen Integrator
- *truncflowanho.m*: Hauptprogramm zur Lösung der Flussgleichung
- Ordner *Minv*: Enthält die berechneten inverse Matrizen als *comma-separated values* (csv)
- Ordner *Output*: Dorthin schreibt MATLAB die Ausgabedateien
- Ordner *SLAC*: Enthält den Quelltext *Hmatrix.m* zur Berechnung der HAMILTONSche Matrix und die benötigte Subroutine *pot.m*
- Ordner *Stuetzstellen*: Enthält zeilenweise die berechneten Kollokationspunkte

Die Ordner *Minv*, *Output* & *Stuetzstellen* müssen für die ordnungsgemäße Ausführung unter dem gleichen Pfad (mit Schreibrechten) wie die m-Dateien liegen.

Quelltextausschnitt 1: *create_basis.m*

```
1 function [rho,rhmax,q,M,Minv,D,Z]=create_basis(basis,N,qmax)
2 if strcmp(basis,'cheb') % Chebychev Polynome
3     % Chebyshev-\textsc{Lobatto} Stützstellen berechnen
4     rho=cos(pi*(0:N)/N); % rho element [-1,1]
5     rhmax=max(rho);
6     q=qmax*sqrt(0.5*(rho+1)); % q element [0,qmax]
7     % Matrix M zur Umrechnung a -> u
8     M=cos((0:N)'*(0:N)*pi/N);
9     % inverse Matrix M zur Umrechnung u -> a // wird für 'cheb' mit MATLAB genau berechnet
10    Minv=inv(M);
11    % Matrix zur Berechnung der Koeffizienten der Ableitung a -> atilde // muss von links angewandt werden
12    D=zeros(N+1,N+1);
13    for n=1:2:N
14        D=D+diag([n,2*(n+1):2:2*N],n);
15    end
16    % Matrix des Zählers der Flussgleichung, d.h. zweite q-Ableitung
17    Z = 4/qmax^2 * ( M*D + 2*diag(rho+1)*M*D*D );
18
19 elseif strcmp(basis,'leg') % Legendre Polynome
20     % Legendre-\textsc{Lobatto} Stützstellen berechnen // da roots(.) nicht besonders genau -->
    Mathematica-Lsg.
```

```

21 | rho=importdata(['.\Stuetzstellen\' , num2str(N),basis,'.txt']); % rho element [-1,1] // Elemente bereits
    |     absteigend sortiert
22 | rhomax=max(rho);
23 | q=qmax*sqrt(0.5*(rho+1)); % q element [0,qmax]
24 | % Matrix M zur Umrechnung a -> u
25 | M=zeros(N+1,N+1);
26 | for n=0:N
27 |     % berechne zuerst die Ergebnisse x^n ... x^0 für die Stützstellen rho
28 |     rhoi=zeros(N+1,n+1);
29 |     for m=n:-1:0
30 |         rhoi(:, n-m+1)=rho.^m;
31 |     end
32 |     M(:,n+1)=rhoi*Legendre(n)';
33 | end
34 | % inverse Matrix M zur Umrechnung u -> a // mit Mathematica berechnet
35 | Minv=importdata(['.\Minv\' , num2str(N),basis,'.txt']);
36 | % Matrix zur Berechnung der Koeffizienten der Ableitung a -> atilde
37 | D=zeros(N+1,N+1);
38 | for n=1:2:N
39 |     D=D+diag(1:2:2*(N-n)+1,n);
40 | end
41 | % Matrix des Zählers der Flussgleichung, d.h. zweite q-Ableitung
42 | Z = 4/qmax^2 * ( M*D + 2*diag(rho+1)*M*D*D );
43 |
44 | elseif strcmp(basis,'her') % Hermite Polynome
45 |     % Stützstellen (Nullstellen des Hermite Polynoms vom Grad N+1) einlesen
46 |     rho=importdata(['.\Stuetzstellen\' , num2str(N),basis,'.txt']); % rho element [-1,1] // Elemente bereits
    |         absteigend sortiert und normiert
47 |     rhomax=max(rho);
48 |     q=qmax*rho; % q element [-qmax,qmax]
49 |     % Matrix M zur Umrechnung a -> u
50 |     M=zeros(N+1,N+1);
51 |     for n=0:N
52 |         rhoi=zeros(N+1,n+1);
53 |         for m=n:-1:0
54 |             rhoi(:, n-m+1)=rho.^m;
55 |         end
56 |         M(:,n+1)=rhoi*Hermite(n)';
57 |     end
58 |     % inverse Matrix M zur Umrechnung u -> a // mit Mathematica berechnet
59 |     Minv=importdata(['.\Minv\' , num2str(N),basis,'.txt']);
60 |     % Matrix zur Berechnung der Koeffizienten der Ableitung a -> atilde
61 |     D=diag(2:2:2*N,1); % diag(.,1) = obere Nebendiagonale
62 |     % Matrix des Zählers der Flussgleichung, d.h. zweite q-Ableitung // andere Substitution beachten
63 |     Z = 1/qmax^2 *M*D*D ;
64 |
65 | elseif strcmp(basis,'lag') % Laguerre Polynome
66 |     % Stützstellen (Nullstellen des Laguerre Polynoms vom Grad N+1) einlesen
67 |     rho=importdata(['.\Stuetzstellen\' , num2str(N),basis,'unskaliert.txt' ]); % rho element [0,rhomax] //
    |         Elemente bereits absteigend sortiert
68 |     rhomax=max(rho);
69 |     q=qmax*sqrt(rho/rhomax); % q element [0,qmax]
70 |     % Matrix M zur Umrechnung a -> u
71 |     M=zeros(N+1,N+1);
72 |     for n=0:N
73 |         rhoi=zeros(N+1,n+1);
74 |         for m=n:-1:0
75 |             rhoi(:, n-m+1)=rho.^m;
76 |         end
77 |         M(:,n+1)=rhoi*Laguerre(n)';
78 |     end
79 |     % inverse Matrix M zur Umrechnung u -> a // mit Mathematica berechnet
80 |     Minv=importdata(['.\Minv\' , num2str(N),basis,'unskaliert.txt' ]);

```

```

81 | % Matrix zur Berechnung der Koeffizienten der Ableitung a -> atilde
82 | D=-1*triu(ones(N+1,N+1),1); % obere Dreiecksmatrix
83 | % Matrix des Zählers der Flussgleichung, d.h. zweite q-Ableitung // andere Substitution beachten
84 | Z= 2*rhomax/qmax^2 * ( M*D + 2*diag(rho)*M*D*D );
85 | else
86 |     [...] % Warnung bei Falsch-Eingabe & Beenden der Routine
87 | end

```

Quelltextausschnitt 2: Hermite.m

```

1 | function f=Hermite(n)
2 | % berechnet n-tes Hermite Polynom a_n x^n + ... + a_0
3 | % Rückgabe als Koeffizienten [a_n ... a_0]
4 | % kann direkt zur Nullstellensuche mittels der Funktion roots genutzt werden
5 | f=zeros(1,n+1);
6 | for k=0:floor(n/2)
7 |     f(2*k+1)=((-1)^k*2^(n-2*k))/(factorial(k)*factorial(n-2*k)); % nach Gleichung (3.13)
8 | end
9 | f=factorial(n)*f;
10 | end

```

Quelltextausschnitt 3: index.m

```

1 | function out=index(in1,in2,Nk)
2 | % Nk+1...Anzahl Spalten / Spaltenindex: 1,...,Nk+1
3 | % in1, in2, out >= 1
4 | out=(Nk+1)*(in1-1)+(in2-1)+1;
5 | end

```

Quelltextausschnitt 4: indices.m

```

1 | function [out1,out2]=indices(in,Nk)
2 | out1=fix((in-1)/(Nk+1))+1; % entspr. ganzzahlige Division // fix...rundet in Richtung Null
3 | out2=mod(in-1,Nk+1)+1;
4 | end

```

Quelltextausschnitt 5: Laguerre.m

```

1 | function f=Laguerre(n)
2 | f=zeros(1,n+1);
3 | for k=0:n
4 |     f(n-k+1)=nchoosek(n,k)*(-1)^k/factorial(k); %nchoosek berechnet Binomialkoeff., nach Gleichung (3.14)
5 | end
6 | end

```

Quelltextausschnitt 6: Legendre.m

```

1 | function f=Legendre(n)
2 | % nicht zu verwechseln mit der Matlab-eigener Funktion legendre !
3 | f=zeros(1,n+1);
4 | for k=0:floor(n/2)
5 |     f(2*k+1)=(-1)^k*factorial(2*n-2*k)/(factorial(n-k)*factorial(n-2*k)*factorial(k)*2^n); %nach Gleichung
6 |     (3.12)
7 | end
8 | end

```

Quelltextausschnitt 7: RHS.m

```

1 | function adot=RHS(k,a,Z,Minv)

```

```

2 | % berechnet RHS für ODE45-Löser
3 | zaehler=Z*a;
4 | adot=Minv*(-1/pi*(zaehler./(k*k+zaehler))); % nach Gleichungen (3.21) & (3.22)
5 | end

```

Quelltextausschnitt 8: *spectralintegration.m*

```

1 | function [A,k,iter]=spectralintegration(qmax,rhmax,kmax,kmin,Mq,Mk,Nq,Nk,rho,kappa,Dq,basis,aStart,tol,
2 |     safety)
3 | if kmax==Inf % Kompaktifizierung
4 |     k=2./(1-kappa)-1; % k element [0,Infinity)
5 |     kfaktor=(1-kappa).^2./2; % substitutionsabhängiger Vorfaktor der k-Differenziation
6 | else % Patches
7 |     kspan=kmax-kmin;
8 |     k=0.5*(kappa*kspan+kmin+kmax); % k element [kmin,kmax]
9 |     kfaktor=ones(length(kappa),1)*2/kspan; % substitutionsabhängiger Vorfaktor der k-Differenziation
10 | end
11 | % Schematisch: Zähler=vorfaktor*(qfaktor*u'+qqfaktor*u'')
12 | if strcmp(basis,'lag') % Sonderfall --> anderen Substitution bei Laguerre-Basis
13 |     vorfaktor=2*rhmax/qmax^2;
14 |     qfaktor=1;
15 |     qqfaktor=2*rho;
16 | elseif strcmp(basis,'her') % Sonderfall --> anderen Substitution bei Hermite-Basis
17 |     vorfaktor=1/qmax^2;
18 |     qfaktor=0;
19 |     qqfaktor=ones(length(rho));
20 | else % Regelfall
21 |     vorfaktor=4/qmax^2;
22 |     qfaktor=1;
23 |     qqfaktor=2*(rho+1);
24 | end
25 | % Matrix zur Berechnung der Koeffizienten der Ableitung A -> atilde
26 | Dk=zeros(Nk+1,Nk+1);
27 | for n=1:2:Nk
28 |     Dk=Dk+diag([n,2*(n+1):2:2*Nk],-n); % -n statt n im letzten Argument entspr. Transposition
29 | end
30 | % Berechne Start-Funktionswerte
31 | uStart=Mq*aStart;
32 | % berechne initial guess
33 | A=aStart/Mk(1,:);
34 | % spektrale Newton-Iteration
35 | F=zeros((Nq+1)*(Nk+1),1);
36 | J=zeros((Nq+1)*(Nk+1),(Nq+1)*(Nk+1)); % Jacobi-Matrix
37 | res=Inf; % Residuum als Schleifen-Abbruchskriterium
38 | iter=0; % Iterationsindex, maximaler Wert wird in while-Schleife begrenzt
39 | % berechne F(A)=0 // 1x außerhalb der Schleife, ab dem zweiten Durchgang wird F bereits zur
40 |     Residuumsbestimmung berechnet
41 | for n=1:Nq+1
42 |     % Anfangswerte
43 |     F(index(n,1,Nk))=Mq(n,:)*A*Mk(1,:)' - uStart(n);
44 |     ZA= vorfaktor*Mq(n,:)*(qfaktor*Dq*A+qqfaktor(n)*Dq*Dq*A); % Zähler
45 |     % alle anderen Werte
46 |     for m=2:Nk+1
47 |         F(index(n,m,Nk))= kfaktor(m)*Mq(n,:)*A*Dk*Mk(m,:)' + 1/pi * (ZA*Mk(m,:))'/( k(m)^2 + ZA*Mk(
48 |             m,:)' );
49 |     end
50 | end
51 | while (res>tol && iter<100)
52 |     % berechne Jacobi-Matrix J = dF_n / dA_m
53 |     for n=1:(Nq+1)*(Nk+1)
54 |         [n1,n2]=indices(n,Nk);

```

```

54 | if n2==1 % Anfangswerte
55 |     for m=1:(Nq+1)*(Nk+1)
56 |         [m1,m2]=indices(m,Nk);
57 |         Adiff=zeros(Nq+1,Nk+1); % Ableitung der Koeffizientenmatrix A nach aktuellem
58 |             Koeffizienten a_{m1,m2}
59 |         Adiff(m1,m2)=1; % hat nur an dieser Stelle eine 1
60 |         J(n,m)=Mq(n1,:)*Adiff*Mk(n2,:);
61 |     end
62 | else % alle anderen Werte
63 |     N= k(n2)^2 + vorkfaktor*(qfaktor*Mq(n1,:)*Dq*A*Mk(n2,:)+qqfaktor(n1)*Mq(n1,:)*Dq*Dq*A*
64 |         Mk(n2,:)); % Nenner
65 |     for m=1:(Nq+1)*(Nk+1)
66 |         [m1,m2]=indices(m,Nk);
67 |         Adiff=zeros(Nq+1,Nk+1); % Ableitung von A nach aktuellem Koeffizienten a_{m1,m2}
68 |         Adiff(m1,m2)=1; % hat nur an dieser Stelle eine 1
69 |         ZA= vorkfaktor*k(n2)^2*Mq(n1,:)*(qfaktor*Dq*Adiff+qqfaktor(n1)*Dq*Dq*Adiff)*Mk(n2,:);
70 |             % Zähler
71 |         J(n,m)=kfaktor(n2)*Mq(n1,:)*Adiff*Dk*Mk(n2,:) + 1/pi * ZA/(N^2);
72 |     end
73 | end
74 | % Lösen des LGS
75 | % Verbessere optional die Kondition der Matrix J abhängig von deren Kondition
76 | kondition=cond(J);
77 | if (kondition<1e10) % Regelfall
78 |     delta=J\(-F);
79 | elseif (kondition>1e14) % QR-Zerlegung
80 |     [Q,R]=qr(J);
81 |     delta=R\(-Q'*F);
82 | else % 1e10<kondition<1e14 // kleine Verbesserung durch Skalieren aller Zeilen auf 1
83 |     dia=diag(1./max(J,[],2));
84 |     delta=(dia*J)\(-dia*F);
85 | end
86 | Delta=reshape(delta,Nk+1,Nq+1)'; % Umwandlung des Vektors in Matrix
87 | A=A+safety*Delta; % neue Koeffizientenmatrix
88 | % Berechne Residuum beruhend auf den neuen Werten, F(A)=0
89 | for n=1:Nq+1
90 |     % Anfangswerte
91 |     F(index(n,1,Nk))=Mq(n,:)*A*Mk(1,:) - uStart(n);
92 |     % alle anderen Werte
93 |     ZA= vorkfaktor*Mq(n,:)*(qfaktor*Dq*A+qqfaktor(n)*Dq*Dq*A); % Zähler
94 |     for m=2:Nk+1
95 |         F(index(n,m,Nk))= kfaktor(m)*Mq(n,:)*A*Dk*Mk(m,:) + 1/pi * (ZA*Mk(m,:))/( k(m)^2 + ZA*
96 |             Mk(m,:) );
97 |     end
98 | end
99 | res=sqrt( (F'*F)/length(F) ); % skalierte euklidische Norm
100 | end

```

Quelltextausschnitt 9: *truncflowanho.m*

```

1 | function [u,a,E,t, gesiter]=truncflowanho(lambda,Nq,qmax,int,basis)
2 | % V(x)=omega2/2*x^2+lambda/24*x^4, Integration mittels verschiedener Integratoren
3 | % Bsp.-Eingabe: [u,a,E,t,gesiter]=truncflowanho(0,10,1,89,'cheb');
4 |
5 | % Variablen anlegen
6 | cutoff=10000;
7 | omega2=1;
8 | gesiter=0; % gesamte Anzahl aller Iterationen
9 | [...]

```

```

10 % Erzeuge Stützstellen und Matrizen Mq & Dq für getroffene Auswahl
11 [rho,rhox,q,Mq,Minv,Dq,Z]=create_basis(basis,Nq,qmax);
12 % für spektrale Integratoren: Lege kappa-Stützstellen und Matrix Mk an
13 if int >= 100
14     tol=1e-8; % Toleranz des Integrators
15     safety=1.0; % Sicherheitsfaktor 0<...<=1 // gedämpftes Newtonverfahren (langsamer, aber sicherer)
16     Nk=10; % Nk+1=Anzahl der Stützstellen in einem Patch, Nk=Grad des höchsten Polynoms
17     % für Variante mit Patches
18     kpatch=[cutoff,0.3*cutoff,0.01*cutoff,0.004*cutoff,0.0015*cutoff,0.0008*cutoff,0.0005*cutoff,0.0002*
19         cutoff,0.0001*cutoff,0]; % Skalen, an denen sich die Patches berühren
19     % Erzeuge Stützstellen und Matrix Mk // k-Integration stets mit Chebyshev
20     % Chebyshev- \textsc{Lobatto} Stützstellen berechnen
21     kappa=cos(pi*(0:Nk)/Nk)'; % kappa element [-1,1]
22     % Matrix Mk zur Umrechnung A -> u
23     Mk=cos((0:Nk)*(0:Nk)*pi/Nk);
24 end
25 % Berechnung der Anfangskoeffizienten durch aLambda=Mq^(-1)*uLambda
26 uLambda=omega2/2*q.^2+lambdax/24*q.^4;
27 aLambda=Minv*uLambda; % Gleichung mit 'Mq\'' zu lösen ist schneller & genauer als inverse Matrix Minv zu
    verwenden // andererseits ist mit Mathematica berechnet inverse numerisch exakt und spart Rechenzeit,
    insb. bei öfterer Anwendung
28 [...] % Bereinige numerische Nullen aufgrund Interpolation
29
30 % k-Evolution Lambda -> 0
31 tic % Start der Zeitmessung
32 if int==0 % 0...expliziter Euler
33     [...]
34 elseif int==4 % 4...Runge-Kutta 4. Ordnung
35     [...]
36 elseif (int >= 10 && int <= 99) % ...integrierter ODE45-Löser
37     Nk=100000; % Anzahl der Stützstellen entlang k-Achse
38     k=linspace(cutoff,0,Nk); % k = 10'000, ... ,0
39     RHSpara=@(t,a) RHS(t,a,Z,Minv); % anonyme Funktion, um RHS die Matrizen Z & Minv zu übergeben
40     options = odeset('RelTol',10^-(floor(int/10)), 'AbsTol',10^-(mod(int,10)), 'Stats', 'on'); % 'RelTol', 1.
        Ziffer von 'int', 'AbsTol', 2. Ziffer von 'int'
41     [kges,ages]=ode45(RHSpara,k,aLambda,options);
42 elseif int==100 % spektraler Integrator mit Patches
43     patchanz=length(kpatch)-1; % Anzahl der Patches
44     aStart=aLambda; % Startwerte belegen
45     ages=zeros(patchanz*(Nk+1),Nq+1); % gesamte Koeffizienten an den Skalenstützstellen
46     kges=zeros(patchanz*(Nk+1),1); % gesamte Skalenstützstellen
47     % Patching-Schleife
48     for n=1:patchanz
49         % Aufruf der Subroutine zur Integration des aktuellen Patches
50         [A,k,iter]=spectralintegration(qmax,rhox,kpatch(n),kpatch(n+1),Mq,Mk,Nq,Nk,rho,kappa,Dq,basis
51             ,aStart,tol,safety);
52         gesiter=gesiter+iter;
53         ages((n-1)*(Nk+1)+1:n*(Nk+1),:)=(A*Mk)';
54         kges((n-1)*(Nk+1)+1:n*(Nk+1))=k;
55         aStart=A*Mk(end,:); % neue Startwerte für nächste Iteration
56     end
57 elseif int==200 % spektraler Integrator mit Kompaktifizierung
58     [A,kges,gesiter]=spectralintegration(qmax,rhox,Inf,0,Mq,Mk,Nq,Nk,rho,kappa,Dq,basis,aLambda,tol,
59         safety);
60     ages=(A*Mk)';
61 else
62     [...] % Warnung bei Falsch-Eingabe & Beenden der Routine
63 end
64 t=toc; % Stopp der Zeitmessung
65
66 % Umrechnung der End-Koeffizienten in Funktionswerte und Interpolation
67 u=Mq*ages(end,:);
68 qq=0:.01:qmax; % q-Werte für Plot der Anfangsverteilung

```

```

67 | [...]
68 | uu=zeros(size(qq));
69 | if strcmp(basis,'cheb')
70 |     for j=0:Nq
71 |         uu=uu+ages(end,j+1)*cos(j*acos(2/qmax^2*qq.^2-1)); % nach Gleichung (3.11)
72 |     end
73 |     u2strich0=Z(Nq+1,:)*ages(end,:); % zur Energieberechnung 2. Ableitung im Ursprung benötigt // Z
       |         enthält Mq bereits
74 | elseif strcmp(basis,'leg')
75 |     for j=0:Nq
76 |         % berechne zuerst die Ergebnisse  $x^n \dots x^0$  für die Plotstellen  $\rho=2*(q/qmax)^{2-1}$ 
77 |         qi=zeros(length(qq),j+1);
78 |         for m=j:-1:0
79 |             qi(:,j-m+1)=(2/qmax^2*qq.^2-1).^m;
80 |         end
81 |         uu=uu+(ages(end,j+1)*qi*Legendre(j)');
82 |     end
83 |     u2strich0=Z(Nq+1,:)*ages(end,:);
84 | elseif strcmp(basis,'her')
85 |     for j=0:Nq
86 |         % berechne zuerst die Ergebnisse  $x^n \dots x^0$  für die Plotstellen  $\rho=q/qmax$ 
87 |         qi=zeros(length(qq),j+1);
88 |         for m=j:-1:0
89 |             qi(:,j-m+1)=(qq/qmax).^m;
90 |         end
91 |         uu=uu+(ages(end,j+1)*qi*Hermite(j)');
92 |     end
93 |     % bei Hermite anderer Weg für 2. Ableitung im Ursprung notwendig, da dort nur bei ungeraden N eine
       |         Stützstelle sitzt
94 |     % berechne Vektor der Funktionswerte  $H_n(0)$ 
95 |     Hn0=[1,zeros(1,Nq)];
96 |     for n=2:Nq
97 |         Hn0(n+1)=2*(1-n)*Hn0(n-1);
98 |     end
99 |     u2strich0=1/qmax^2*Hn0*Dq*Dq*ages(end,:); % zur Energieberechnung zweite Ableitung im Ursprung
       |         benötigt // Z enthält Mq bereits
100 | elseif strcmp(basis,'lag')
101 |     [...]
102 | end
103 | % Berechnung der Energie
104 | E=zeros(2,1);
105 | E(1)=min(uu);
106 | E(2)=min(uu)+sqrt(u2strich0);
107 | % Darstellung / Speichern der Lösung
108 | [...]
109 | end

```

B. Literatur

- [1] A. Wipf. *Statistical approach to quantum field theory. An introduction*. Lecture Notes in Physics. Springer Berlin Heidelberg, 2013.
- [2] C. Bagnuls und C. Bervillier. „Exact renormalization group equations. An Introductory review“. In: *Phys.Rept.* 348 (2001), S. 91. arXiv: hep-th/0002034 [hep-th].
- [3] J. Berges, N. Tetradis und C. Wetterich. „Nonperturbative renormalization flow in quantum field theory and statistical physics“. In: *Phys.Rept.* 363 (2002), S. 223–386. arXiv: hep-ph/0005122 [hep-ph].
- [4] J. Polonyi. „Lectures on the functional renormalization group method“. In: *Central Eur.J.Phys.* 1 (2003), S. 1–71. arXiv: hep-th/0110026 [hep-th].
- [5] H. Gies. „Introduction to the functional RG and applications to gauge theories“. In: *Lect.Notes Phys.* 852 (2012), S. 287–348. arXiv: hep-ph/0611146 [hep-ph].
- [6] C. Wetterich. „Exact evolution equation for the effective potential“. In: *Phys.Lett.* B301 (1993), S. 90–94.
- [7] R. Hedden u. a. „A functional renormalization group approach to zero-dimensional interacting systems“. In: *Journal of Physics Condensed Matter* 16 (Juli 2004), S. 5279–5296.
- [8] D. F. Litim. „Optimized renormalization group flows“. In: *Phys.Rev.* D64 (2001), S. 105007. arXiv: hep-th/0103195 [hep-th].
- [9] J. P. Boyd. *Chebyshev and Fourier spectral methods*. Second Edition. Courier Dover Publications, 2000.
- [10] L. N. Trefethen. *Spectral Methods in MATLAB*. Society for Industrial und Applied Mathematics, 2000. eprint: <http://epubs.siam.org/doi/pdf/10.1137/1.9780898719598>.
- [11] D. Gottlieb und S. A. Orszag. *Numerical Analysis of Spectral Methods*. Society for Industrial und Applied Mathematics, 1977. eprint: <http://epubs.siam.org/doi/pdf/10.1137/1.9781611970425>.
- [12] I. N. Bronstein u. a. *Taschenbuch der Mathematik*. 7th revised and enlarged ed. Frankfurt am Main: Harri Deutsch, 2008.
- [13] E. W. Weisstein. *Legendre Polynomial*. MathWorld - A Wolfram Web Resource. Stand: 30.08.20.14 18:00. URL: <http://mathworld.wolfram.com/LegendrePolynomial.html>.
- [14] M. Abramowitz und I. A. Stegun, Hrsg. *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*. New York: Dover Publications, Inc., 1965.
- [15] J. Hennig und M. Ansorg. „A Fully Pseudospectral Scheme for Solving Singular Hyperbolic Equations on Conformally Compactified Space-Times“. In: *J.Hyperbol.Diff.Equat.* 6 (2009), S. 161. arXiv: 0801.1455 [gr-qc].
- [16] G. Ierley, B. Spencer und R. Worthing. „Spectral methods in time for a class of parabolic partial differential equations“. In: *Journal of Computational Physics* 102.1 (1992), S. 88–97.
- [17] J. Förster, A. Saenz und U. Wolff. „Matrix algorithm for solving Schrödinger equations with position-dependent mass or complex optical potentials“. In: *Phys. Rev. E* 86 (1 2012), S. 016701. URL: <http://link.aps.org/doi/10.1103/PhysRevE.86.016701>.
- [18] A. Kirchberg, J. Lange und A. Wipf. „From the Dirac operator to Wess-Zumino models on spatial lattices“. In: *Annals Phys.* 316 (2005), S. 357–392. arXiv: hep-th/0407207 [hep-th].
- [19] A. Wipf. *Quantenmechanik I*. Vorlesungsskript. Jena, 2005.

- [20] D. Zappala. „Improving the renormalization group approach to the quantum mechanical double well potential“. In: *Phys.Lett.* A290 (2001), S. 35–40. arXiv: [quant-ph/0108019](https://arxiv.org/abs/quant-ph/0108019) [quant-ph].

Für diese Arbeit wurde folgende Software inklusive ihrer Dokumentationen genutzt:

- MATLAB R2009b
- Wolfram Mathematica 8

Danksagung

Als Erstes gilt mein Dank Prof. Wipf für die Bereitstellung dieser interessanten Bachelorarbeit sowie deren Betreuung. Bei Prof. Ammon möchte ich mich für die Übernahme des Zweitgutachtens bedanken. Von ganzem Herzen danke ich Benjamin für die hervorragende Betreuung. Ich fühlte mich jederzeit willkommen und habe durch unsere Gespräche viel gelernt. Mein abschließender Dank geht an die Korrekturleser Markus, Maximilian und Sebastian.

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Abschlussarbeit selbstständig und nur mit den angegebenen Hilfsmitteln und Quellen verfasst habe.

Ich erkläre ausdrücklich, dass ich sämtliche in der Arbeit verwendeten fremden Quellen, auch aus dem Internet, als solche kenntlich gemacht habe. Insbesondere bestätige ich, dass ich ausnahmslos sowohl bei wörtlich übernommenen Aussagen bzw. unverändert übernommenen Tabellen, Grafiken u. Ä. (Zitaten) als auch bei in eigenen Worten wiedergegebenen Aussagen bzw. von mir abgewandelten Tabellen, Grafiken u. Ä. anderer Autorinnen und Autoren (indirektes Zitieren) die Quelle angegeben habe. Die Arbeit wurde in gleicher oder ähnlicher Form bisher bei keiner anderen Institution eingereicht.

Seitens des Verfassers bestehen keine Einwände, die vorliegende Arbeit für die öffentliche Nutzung in der Thüringer Universitäts- und Landesbibliothek zur Verfügung zu stellen.

Jena, den 19.09.2014
