



Universität Bremen



Institut für Theoretische Physik

Numerische Behandlung von Integro-Differentialgleichungen aus der Beschreibung von Polaronen

Diplomarbeit

Nikolaj Kemlein

Betreuer und Gutachter:

Prof. Dr. Frank Jahnke

Prof. Dr. Alfred Schmidt

Inhaltsverzeichnis

1	Einleitung	2
1.1	Begriffsklärung	2
1.2	Problematik	3
2	Theoretische Grundlagen	4
2.1	Elektronen	4
2.2	Gitterschwingungen	5
2.3	Elektron-Phonon-Wechselwirkung	7
2.4	Gesamtsystem und Heisenberg-Bewegungsgleichungen	8
3	Beschreibung der Elektron-Phonon-Wechselwirkung	9
3.1	Greensfunktionen	9
3.2	Bewegungsgleichungen für retardierte Greensfunktionen	10
3.3	Umformungen der Dyson-Gleichung	14
4	Andere Lösungsansätze	16
5	Adams - Bashforth - Moulton	18
5.1	Allgemeine Idee	18
5.2	Polynomiale Interpolation	18
5.3	ABM113	19
6	Modifikationen des ABM113	22
6.1	Aufbau der Interpolationspolynome	22
6.2	Organisation des Vergangenheitspeichers	24
6.3	Multiplikation und Integration, Variante eins	26
6.4	Multiplikation und Integration, Variante zwei	28
7	Ergebnisse	32
7.1	Konstanter Memory-Kern	32
7.2	Linearer Memory-Kern	33
7.3	Physikalischer Memory-Kern: Polaron-Selbstenergie	35
8	Zusammenfassung und Ausblick	43
	Literatur	48

1 Einleitung

In der Quantenkinetik werden die Wechselwirkungen zwischen Ladungsträgern und Halbleiterkristallen untersucht, unter anderem auch die Spektralfunktionen von Polaronen. Zur Berechnung dieser Spektralfunktionen wird die Dyson-Gleichung gelöst, welche eine Integro-Differentialgleichung mit nicht linearem Memory-Kern ist. Diese Art von Gleichungen lassen sich analytisch nicht lösen, deswegen ist die numerische Berechnung unumgänglich. Bisherige Verfahren lösen die Dyson-Gleichung zwar, brauchen aber teilweise sehr lange und lassen vor allem keine verlässlichen Aussagen über die (lokale) Genauigkeit der Lösung zu, was für die weiterverwendung, zum Beispiel zur Berechnung von Streuraten, sehr hilfreich wäre. Ziel dieser Arbeit ist es, ein Verfahren zu konstruieren, das einerseits möglichst schnell ist und andererseits eine lokale Genauigkeit einhält.

Zu diesem Zweck wird, nachdem in Kapitel 2 die physikalischen Grundlagen erläutert wurden, die benötigt werden um in Kapitel 3 die Dyson-Gleichung herzuleiten, in Kapitel 5 und 6 ein Verfahren entwickelt, das eine adaptive Schrittweitensteuerung und eine effektive und fehler technisch kalkulierbare Auswertung des Differentials verwendet. Schließlich wird die Leistungsfähigkeit dieses Verfahrens in Kapitel 7 getestet und bewertet.

1.1 Begriffsklärung

Integro-Differentialgleichungen sind Differentialgleichungen, deren Differential ein Integral enthält:

$$\begin{aligned} y &: \mathbb{R} \rightarrow \mathbb{C}^n \\ B &: \mathbb{R} \times \mathbb{C}^n \rightarrow \mathbb{C}^n \\ \frac{d}{dt}y(t) &= \int_0^t B(t', y(t')) \cdot y(t') dt'. \end{aligned} \quad (1.1)$$

Gleichungen dieser Art heißen Volterra-Gleichungen [6] und lassen sich oft in Differentialgleichungen zweiter Ordnung umformen, indem die ganze Gleichung differenziert und anschließend der Hauptsatz der Differential- und Integralrechnung auf das Differential angewendet wird. Diese Umformung ist nur dann möglich, wenn der Kern des Integrals $B(t - t', y(t - t'))$ nicht von der gesuchten Funktion zum Zeitpunkt t , $y(t)$ abhängt; Gleichungen der Form

$$\begin{aligned} y &: \mathbb{R} \rightarrow \mathbb{C}^n \\ B &: \mathbb{R} \times \mathbb{R} \times \mathbb{C}^n \times \mathbb{C}^n \rightarrow \mathbb{C}^n \\ \frac{d}{dt}y(t) &= \int_0^t B(t, t', y(t), y(t')) \cdot y(t') dt' \end{aligned} \quad (1.2)$$

lassen sich im Allgemeinen nicht mehr in eine reine Differentialgleichung überführen und bedürfen daher einer besonderen Behandlung. Die Kerne solcher Integrale nennt man in der Physik Gedächtnis- oder Memory-Kerne, da die zeitliche Kausalität sicherstellt, dass die Zeitpunkte t und t' , die in $B(t, t', y(t), y(t'))$ vorkommen, in der Vergangenheit liegen. In dieser Arbeit werden nur Memory-Kerne behandelt, die von einer Zeitdifferenz $(t - t')$ und $y(t - t')$ abhängen. Mit der Anfangsbedingung $y(0) = 1$ ergibt sich das grundlegende Anfangswertproblem für diese Arbeit:

$$\begin{aligned} y &: \mathbb{R} \rightarrow \mathbb{C}^n \\ B &: \mathbb{R} \times \mathbb{C}^n \rightarrow \mathbb{C}^n \\ \frac{d}{dt}y(t) &= \int_0^t B(t - t', y(t - t')) \cdot y(t') dt' := w(t, y(t)) \\ y(0) &= 1. \end{aligned} \quad (1.3)$$

1.2 Problematik

Bei der numerischen Behandlung von Integro-Differentialgleichungen mit Memory-Kern ergibt sich ein Hauptproblem: In jedem Schritt, den die Differentialgleichung voranschreitet, ändert sich der Kern des Integrals und das Integral muss bis zu dem aktuellen t_n immer wieder vollständig neu gelöst werden. Das führt dazu, dass jeder weitere Schritt in der Differentialgleichung länger dauert als die vorherigen, zumindest wenn man eine konstante Schrittweite zu Grunde legt. Dies lässt sich an einem Beispiel verdeutlichen:

Zeitindex	Auswertung der rechten Seite	
1	$\int_{t_0}^{t_1} B_1 y_1$	
2	$\int_{t_0}^{t_1} B_2 y_1 + \int_{t_1}^{t_2} B_1 y_2$	(1.4)
3	$\int_{t_0}^{t_1} B_3 y_1 + \int_{t_1}^{t_2} B_2 y_2 + \int_{t_2}^{t_3} B_1 y_3$	
4	$\int_{t_0}^{t_1} B_4 y_1 + \int_{t_1}^{t_2} B_3 y_2 + \int_{t_2}^{t_3} B_2 y_3 + \int_{t_3}^{t_4} B_1 y_4.$	

Dabei sind die Integrale diskret zu verstehen und die B_n und y_n die Diskretisierungen der oben verwendeten Größen $B(t - t', y(t - t'))$ und $y(t')$. In dem Beispiel wird deutlich, dass in jedem weiteren Schritt der Differentialgleichung die Integration immer auf einem zusätzlichen Intervall durchgeführt werden muss und die bereits berechneten Integrale der Vergangenheit nicht weiterverwendet werden können. Die Änderung des Zeitaufwandes steigt linear mit jedem Schritt, und der Zeitaufwand der Lösung der Differentialgleichung wächst somit quadratisch mit der Anzahl der Punkte des zu Grunde gelegten Gitters. Es bietet sich also an, ein Verfahren zu verwenden, das ein adaptives Gitter verwendet.

2 Theoretische Grundlagen

Integro-Differentialgleichungen werden benötigt, um spektrale Eigenschaften von Polaronen zu berechnen. Polaronen sind ein physikalisches Konzept, das die Wechselwirkung von Ladungsträgern (Elektronen) mit den in einem (Halbleiter-) Kristall vorhandenen Gitterschwingungen beschreibt. Dabei wird angenommen, dass keine zusätzlichen Ladungsträger angeregt werden können, mit denen ein betrachtetes Polaron wechselwirkt und das die Gitterschwingungen ein Bad im thermischen Gleichgewicht darstellen. Das Konzept der Polaronen wird in dieser Arbeit an einer Halbleiterstruktur aus einem InGaAs-Quantenpunkt auf einer InGaAs-Benetzungsschicht (Wetting-Layer) untersucht, die von GaAs umgeben ist. Aufgrund der kleinen Bandlücken von InGaAs ist in dieser Struktur das Potential für Ladungsträger zur Umgebung abgesenkt, so dass sich lokalisierte Zustände bilden, die in den Quantenpunkt in alle Raumrichtungen und im Wetting-Layer in einer Raumrichtung eingeschlossen sind. Dadurch sind die gebundenen Zustände im Quantenpunkt diskret und die Zustände des Wetting-Layer in zwei Dimensionen kontinuierlich. Die diskreten Zustände im Quantenpunkt werden in Analogie zur Atomstruktur s- und p-Schale genannt. In Abbildung 1 ist die Struktur einer solcher Kristallstruktur schematisch dargestellt. In Kapitel 2 und 3 wird vorerst nur der Wetting-Layer untersucht, da dieser ein homogenes System und somit etwas einfacher zu betrachten ist.

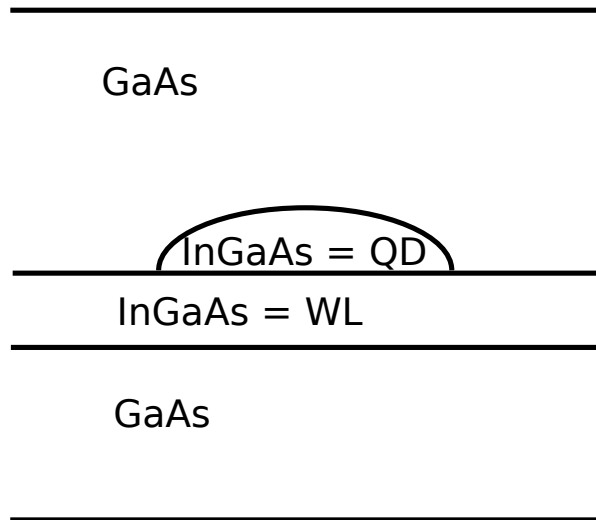


Abbildung 1: Darstellung eines Quantenpunktes und eines Wetting-Layer aus InGaAs in einer GaAs Umgebung [9].

In dem folgenden Kapitel werden die Grundlagen für die Herleitung der Dyson-Gleichung, die die Polaronen in der betrachteten Halbleiterstruktur beschreibt, kurz erläutert und anschließend in Kapitel 3 eine Herleitung dieser Gleichung beschrieben.

2.1 Elektronen

In zweiter Quantisierung werden für die Feldoperatoren freier Elektronen im Kristallgitter, $\Psi(r)$, Vertauschungsrelationen mit kanonisch konjugierten Variablen $\Psi^\dagger(r)$

$$[\Psi(r), \Psi^\dagger(r')] = \delta(r - r') \quad (2.1)$$

und der Hamiltonoperator

$$H = \int \Psi^\dagger(r) \left\{ \frac{-\hbar^2}{2m} \Delta + U(r) \right\} \Psi(r) dr \quad (2.2)$$

aufgestellt. Als Nächstes werden die Feldoperatoren in der Bloch-Basis $\Phi_k(r)$, $\Phi_k^*(r)$

$$\Psi(r) = \sum_k a_k \Phi_k(r) \text{ und } \Psi^\dagger(r) = \sum_k a_k^\dagger \Phi_k^*(r) \quad (2.3)$$

entwickelt, in der der Hamiltonoperator H diagonal ist. Aus der Orthonormiertheit der Basisfunktionen folgen die Antikommutatorrelationen für elektronische Vernichter- und Erzeuger-Operatoren

$$\begin{aligned} [a_k, a_{k'}^\dagger]_+ &= \delta_{kk'} \\ [a_k, a_{k'}]_+ &= [a_k^\dagger, a_{k'}^\dagger]_+ = 0 \end{aligned} \quad (2.4)$$

und es ergibt sich der Hamiltonoperator für freie Elektronen im Kristallgitter in zweiter Quantisierung [1], in welcher Vielteilchen-Wechselwirkungen besser zu behandeln sind:

$$H = \sum_k \varepsilon_k a_k^\dagger a_k. \quad (2.5)$$

Dabei ist ε_k die wechselwirkungsfreie Energie eines Elektrons im Bloch-Zustand k . In der Effektivmassennäherung [1] erhält man die Dispersion

$$\varepsilon_k = \frac{\hbar^2 k^2}{2\mu} + E_G. \quad (2.6)$$

mit dem Impuls $\hbar k$ der Elektronen im Zustand k , der Bandlücke E_G und der über

$$\frac{1}{\mu} = \frac{1}{\hbar^2} \frac{\partial^2 \varepsilon_k}{\partial |k|^2} \quad (2.7)$$

definierten effektiven Masse μ .

2.2 Gitterschwingungen

- Klassische Beschreibung

Viele wichtige Eigenschaften von Vielteilchensystemen lassen sich durch die Dynamik der Atomkerne beschreiben. Im Rahmen der Born-Oppenheimer-Näherung ist diese Dynamik unabhängig von der Dynamik der Elektronen, und für hinreichend kleine Auslenkungen aus der Gitterposition kann die Dynamik der Atomkerne als System gekoppelter harmonischer Oszillatoren beschrieben werden. Für linear abwechselnd angeordnete Atome der Sorte A mit Masse m_A und B mit Masse m_B , die sich im Abstand d zu einander befinden, lässt sich für eine harmonische Federkonstante C , mit der jedes Atom an seinen nächsten Nachbarn gekoppelt ist, folgendes System von Bewegungsgleichungen aufstellen [3]:

$$\begin{aligned} m_A \frac{d^2 x_n^A(t)}{dt^2} &= C(x_{n-1}^B(t) + x_{n+1}^B(t) - 2x_n^A(t)) \\ m_B \frac{d^2 x_{n+1}^B(t)}{dt^2} &= C(x_{n+2}^A(t) + x_n^A(t) - 2x_{n+1}^B(t)). \end{aligned} \quad (2.8)$$

Wenn man für $x_n^A(t)$ und $x_{n+1}^B(t)$ ebene Wellen

$$\begin{aligned} x_n^A(t) &= E_A e^{nqd - \omega t} \\ x_{n+1}^B(t) &= E_B e^{(n+1)qd - \omega t} \end{aligned} \quad (2.9)$$

als Ansatz wählt, folgt die Dispersionsrelation

$$\omega_{1,2}^2(q) = C \frac{m_A + m_B}{m_A m_B} \pm \frac{C}{m_A m_B} \sqrt{m_A^2 + m_B^2 + 2m_A m_B \cos(2qd)}. \quad (2.10)$$

Dabei beschreibt ω_1 Atome, die longitudinal gegenphasig, und ω_2 Atome, die longitudinal in Phase schwingen.

- Quantenmechanische Behandlung

Eine quantenmechanische Betrachtung zeigt, dass die Gitterschwingungen der Atomkerne quantisiert sind, diese gequantelten Gitterschwingungen werden als Phononen bezeichnet. Die gegenphasig schwingenden Atome heißen in der Quantenmechanik optische Phononen, die Atome, die in Phase schwingen, werden akustische Phononen genannt. Die Namen weisen auf die unterschiedlichen Eigenschaften der beiden Arten der Schwingungen hin: Während gegenphasig schwingende Atome an elektrische Felder koppeln, transportieren in Phase schwingende Atome Schallwellen. In dieser Arbeit wird nur die Kopplung von Elektronen an optische Phononen betrachtet. Außerdem werden Halbleitersysteme betrachtet, bei denen sich die angeregten Zustände der Elektronen nahe am Leitungsbandminimum befinden, dadurch sind fast ausschließlich Phononen großer Wellenlänge an der Phonon-Elektron-Wechselwirkung beteiligt [7]. In diesem Grenzfall kann die Dispersion optischer Phononen ω_{LO} als konstant angenommen werden. Dies wird als Einstein-Modell bezeichnet:

$$\omega_{LO} = \frac{E_{LO}}{\hbar} = \text{const.} \quad (2.11)$$

Quantenmechanisch lassen sich Gitterschwingungen (für ausreichend kleine Auslenkungen) als Ensemble von harmonischen Oszillatoren mit dem folgenden Hamiltonoperator beschreiben:

$$H = \sum_q \hbar \omega_{LO} \left(b_q^\dagger b_q + \frac{1}{2} \right). \quad (2.12)$$

Außerdem gelten für die Phononen-Erzeugungs- und Vernichtungsoperatoren die folgenden Kommutatorrelationen

$$\begin{aligned} [b_q, b_{q'}^\dagger] &= \delta_{qq'} \\ [b_q, b_{q'}] &= [b_q^\dagger, b_{q'}^\dagger] = 0. \end{aligned} \quad (2.13)$$

2.3 Elektron-Phonon-Wechselwirkung

Bisher wurde erläutert, wie freie Elektronen und das Kristallgitter separat beschrieben werden können. Nun soll die Wechselwirkung von Elektronen und Phononen untersucht werden [7]. Dafür wird das gitterperiodische Potential eines Festkörpers

$$U(r) = \sum_j u(r - R_j) \quad (2.14)$$

betrachtet, wobei die R_j die Gleichgewichtspositionen der Atomrümpfe und u das Rumpfpotential eines Atoms repräsentieren. Der Hamiltonoperator dieses Systems ist

$$H_0 = \sum_i \left(\frac{p_i^2}{2m} + U(r_i) \right). \quad (2.15)$$

Die Elektron-Phonon-Wechselwirkung wird für kleine Gitterauslenkungen Q_j aus der Ruhelage $R_j^{(0)}$ in harmonischer Näherung beschrieben, also

$$R_j = R_j^{(0)} + Q_j \quad (2.16)$$

und die Rumpfpotentiale linear nach den Auslenkungen entwickelt:

$$U(r) = \sum_j u(r - (R_j^{(0)} + Q_j)) = \sum_j u(r - R_j^{(0)}) - \sum_j Q_j \cdot \nabla u(r - R_j^{(0)}). \quad (2.17)$$

Während sich der erste Term mit dem Operator der kinetischen Energie zu einem gitterperiodischen Hamiltonoperator der Form (2.15) zusammenfassen lässt, führt der zweite Term auf einen Elektron-Phonon-Wechselwirkungshamiltonoperator:

$$H_{e-ph} = \sum_i \underbrace{\left(- \sum_j Q_j \cdot \nabla u(r_i - R_j^{(0)}) \right)}_{=: \tilde{u}(r_i)}. \quad (2.18)$$

Mit Hilfe der Teilchendichte $\rho(r) = \sum_i \delta(r - r_i)$ lässt sich dieser in Integralform schreiben:

$$H_{e-ph} = \int \rho(r) \tilde{u}(r) d^3r. \quad (2.19)$$

Die Auslenkungen Q_j werden durch Entwicklung nach phononischen Erzeuger- und Vernichter-Operatoren, für die gemäß (2.13) die Kommutatorrelationen gelten, quantisiert:

$$Q_j = \sum_q \sqrt{\frac{\hbar}{2\rho\omega_q V}} e_q e^{iq \cdot R_j^{(0)}} [b_q + b_{-q}^\dagger], \quad (2.20)$$

was zusammen mit einer Fourierreihenentwicklung für die Rumpfpotentiale auf den Hamiltonoperator

$$H_{e-ph} = - \sum_q \sqrt{\frac{\hbar}{2\rho\omega_q V}} iqu_q \int e^{iq \cdot r} \rho(r) [b_q + b_{-q}^\dagger] d^3r \quad (2.21)$$

führt. In zweiter Quantisierung lässt sich die Teilchendichte $\rho(r)$ als

$$\rho(r) = \Psi^\dagger(r)\Psi(r) \quad (2.22)$$

darstellen. Zusammen mit der Basisentwicklung nach der Bloch-Basis (2.3) ergibt sich

$$\int e^{iq \cdot r} \rho(r) d^3r = \int e^{iq \cdot r} \rho(r) \sum_{k,k'} a_k^\dagger \Phi_k^*(r) a_{k'} \Phi_{k'}(r) d^3r = \sum_{k,k'} \langle k | e^{iq \cdot r} | k' \rangle a_k^\dagger a_{k'}, \quad (2.23)$$

woraus sich dann der Hamiltonoperator für die Phonon-Elektron-Wechselwirkung

$$H_{e-ph} = \sum_{k,k',q} M_{q,k,k'} a_k^\dagger a_{k'} [b_q + b_{-q}^\dagger] \quad (2.24)$$

mit den Kopplungsmatrixelementen

$$M_{q,k,k'} = M_q \langle k | e^{iq \cdot r} | k' \rangle = -i \sqrt{\frac{\hbar}{2\rho\omega_q V}} i q u_q \langle k | e^{iq \cdot r} | k' \rangle \quad (2.25)$$

ergibt. Für das betrachtete System von optischen Phononen sind die Matrixelemente konkret gegeben durch

$$|M_q^{LO}|^2 = \frac{\hbar\omega_{LO}}{2V} \frac{e^2}{\varepsilon_0 q^2} \left(\frac{1}{\varepsilon_\infty} - \frac{1}{\varepsilon} \right) = M_{LO}^2 \frac{1}{q^2}. \quad (2.26)$$

Da ein räumlich homogenes System betrachtet wird, muss wegen der Impulserhaltung außerdem $k' = k + q$ gelten.

2.4 Gesamtsystem und Heisenberg-Bewegungsgleichungen

An dieser Stelle wird der formale Wechsel vom Schrödinger ins Heisenbergbild durchgeführt, dass heißt, dass jetzt nicht mehr die Wellenfunktionen zeitabhängig sind, sondern die Operatoren. Als Hamiltonoperator für das Gesamtsystem ergibt sich dann, wenn man die Zeitabhängigkeit berücksichtigt:

$$\begin{aligned} H(t) &= H_e(t) + H_{ph}(t) + H_{e-ph}(t) \\ &= \sum_k \varepsilon_k a_k^\dagger(t) a_k(t) + \sum_q \hbar\omega_{LO} (b_q^\dagger(t) b_q(t) + \frac{1}{2}) + \sum_{k,q} M_q a_{k+q}^\dagger(t) a_k(t) (b_q(t) + b_{-q}^\dagger(t)). \end{aligned} \quad (2.27)$$

Mit diesem Hamiltonoperator für das Gesamtsystem lassen sich dann Heisenberg-Bewegungsgleichungen für die Vernichter-Operatoren der Elektronen und Phononen aufstellen:

$$i\hbar \frac{d}{dt} a_k(t) = [H(t), a_k(t)] = \varepsilon_k a_k(t) + \sum_q M_q (b_q(t) + b_{-q}^\dagger(t)) a_{k-q}(t), \quad (2.28)$$

$$i\hbar \frac{d}{dt} b_k(t) = [H(t), b_k(t)] = \hbar\omega_{LO} b_k(t) + M_{-q} \sum_k a_k^\dagger(t) a_k(t). \quad (2.29)$$

Die Gleichungen für die Erzeuger-Operatoren ergeben sich analog.

3 Beschreibung der Elektron-Phonon-Wechselwirkung

3.1 Greensfunktionen

In der Quantenmechanik wird die Wahrscheinlichkeit berechnet, ein gewisses Messergebnis zu erhalten. Diese Wahrscheinlichkeit wird über den Erwartungswert des zugehörigen Operators

$$\langle A \rangle \quad (3.1)$$

beschrieben, welcher sich über die Spur des Operators berechnen lässt:

$$\langle A \rangle = \text{Tr}\{\rho A\}. \quad (3.2)$$

Verschiedene Eigenschaften von Elektronen werden über Einteilchenoperatoren beschrieben

$$A(t) = \sum_{s,s'} \int \int \langle r', s' | A | r, s \rangle \Psi_{s'}^\dagger(r', t) \Psi_s(r, t) d^3 r' d^3 r, \quad (3.3)$$

welche sich durch die Feldoperatoren $\Psi_s^\dagger(r, t)$, $\Psi_s(r, t)$ und den Spin s und s' ausdrücken lassen. Der Erwartungswert des Einteilchenoperators lautet

$$\langle A(t) \rangle = \sum_{s,s'} \int \int \langle r', s' | A | r, s \rangle \text{Tr}\{\rho \Psi_{s'}^\dagger(r', t) \Psi_s(r, t)\} d^3 r' d^3 r. \quad (3.4)$$

Es wird also $\langle \Psi_{s'}^\dagger(r', t) \Psi_s(r, t) \rangle$ benötigt, dafür wird die Bewegungsgleichung

$$\frac{d}{dt} \langle \Psi^\dagger \Psi \rangle = T[\langle \Psi^\dagger \Psi \rangle] + V[\langle \Psi^\dagger \Psi^\dagger \Psi \Psi \rangle] \quad (3.5)$$

gelöst, wobei T und V die von Ψ unabhängigen Teile der Gleichung zusammenfassen. Es entstehen Terme der Form $[\langle \Psi^\dagger \Psi^\dagger \Psi \Psi \rangle]$, für die wieder eine Bewegungsgleichung zu lösen ist, die dann einen Term der Form $[\langle \Psi^\dagger \Psi^\dagger \Psi^\dagger \Psi \Psi \rangle]$ beinhaltet und so weiter. Es ergibt sich also eine nicht abbrechende Hierarchie von Bewegungsgleichungen, die an irgendeinem Punkt trunziert werden muss. Eine andere Herangehensweise ist, allgemeinere zweizeitige Erwartungswerte im Grundzustand $|\Phi_G\rangle$ zu betrachten [5]:

$$\begin{aligned} \langle \Phi_G | \Psi_{s'}^\dagger(r', t') \Psi_s(r, t) | \Phi_G \rangle &= -i\hbar G_{ss'}^<(r, t, r', t') \\ \langle \Phi_G | \Psi_{s'}(r', t') \Psi_s^\dagger(r, t) | \Phi_G \rangle &= i\hbar G_{ss'}^>(r, t, r', t') \end{aligned} \quad (3.6)$$

Dabei entspricht $G_{ss'}^>(r, t, r', t')$ der Wahrscheinlichkeitsamplitude zur Zeit t' dafür, dass sich das System im Zustand $\Psi_{s'}^\dagger(r', t') |\Phi_G\rangle$, das heißt ein Elektron mit Spin s' am Ort r' , befindet, wenn zur Zeit t am Ort r ein Elektron mit Spin s zum System hinzugefügt, also $\Psi_s^\dagger(r, t) |\Phi_G\rangle$ erzeugt wurde.

Mit der Konvention $\Psi_{s_1}(r_1, t_1) = \Psi(1)$ ergibt sich die kompaktere Schreibweise

$$\begin{aligned} G^<(1, 1') &= -\frac{i}{\hbar} \langle \Psi^\dagger(1') \Psi(1) \rangle \\ G^>(1, 1') &= \frac{i}{\hbar} \langle \Psi(1) \Psi^\dagger(1') \rangle \end{aligned} \quad (3.7)$$

und damit die zeitgeordnete Greensfunktion:

$$G(1, 1') = \begin{cases} G^<(1, 1') &= -\frac{i}{\hbar} \langle \Psi^\dagger(1') \Psi(1) \rangle & \text{für } t_1 < t'_1 \\ G^>(1, 1') &= \frac{i}{\hbar} \langle \Psi(1) \Psi^\dagger(1') \rangle & \text{für } t_1 > t'_1, \end{cases} \quad (3.8)$$

kompakter geschrieben

$$G(1, 1') = \Theta(t_1 - t'_1) G^>(1, 1') + \Theta(t'_1 - t_1) G^<(1, 1'). \quad (3.9)$$

Da sie besser für Berechnungen geeignet ist und einen einfacheren Zugang zu spektralen Eigenschaften des Systems bietet, wird die retardierte Greensfunktion

$$G^r(1, 1') = \Theta(t_1 - t'_1) [G^>(1, 1') - G^<(1, 1')]. \quad (3.10)$$

eingeführt. In der Bloch-Basis (2.3) für $\Psi(1')$ und $\Psi^\dagger(1)$ ergibt sich die Einteilchen-Greensfunktion für Elektronen

$$\begin{aligned} G_\alpha^r(t, t') &= -\frac{i}{\hbar} \Theta(t - t') \langle a_k(t) a_k^\dagger(t') + a_k^\dagger(t') a_k(t) \rangle \\ &= -\frac{i}{\hbar} \Theta(t - t') \langle [a_k(t), a_k^\dagger(t')]_+ \rangle. \end{aligned} \quad (3.11)$$

Die Greensfunktionen sind auch deshalb eine gute Wahl, weil sie Erwartungswerte von beliebigen Einteilchenoperatoren, den Erwartungswert der Gesamtenergie und das Anregungsspektrum des Vielteilchensystems relativ gut abrufbar enthalten. Beispielsweise werden die spektralen Eigenschaften der Quasiteilchen im System durch

$$\hat{G}_k(\omega) = -\frac{1}{\pi} \text{Im}(\tilde{G}_k^r(\omega)) \quad (3.12)$$

beschrieben, wobei $\tilde{G}_k^r(\omega)$ die Fouriertransformierte von $G_k^r(t)$ ist. An dieser Stelle sind bereits weiter unten erläuterte Näherungen mit berücksichtigt, die die Zweizeitigkeit der Greensfunktion eliminieren.

3.2 Bewegungsgleichungen für retardierte Greensfunktionen

Die Herleitung in diesem Kapitel richtet sich nach [4]. Für die retardierten Greensfunktionen (3.11) lassen sich die folgenden Bewegungsgleichungen aufstellen und umformen:

$$\begin{aligned} i\hbar \frac{\partial}{\partial t} G_k^r(t, t') &= i\hbar \frac{\partial}{\partial t} \left(-\frac{i}{\hbar} \Theta(t - t') \langle [a_k(t), a_k^\dagger(t')]_+ \rangle \right) \\ &= \underbrace{\frac{\partial}{\partial t} \Theta(t - t')}_{=\delta(t-t')} \underbrace{\langle [a_k(t), a_k^\dagger(t')]_+ \rangle}_{=1 \text{ für } t=t'} - \frac{i}{\hbar} \langle [i\hbar \frac{\partial}{\partial t} a_k(t), a_k^\dagger(t')]_+ \rangle. \end{aligned} \quad (3.13)$$

Zum Lösen dieser Bewegungsgleichung bleibt also das Berechnen von $i\hbar \frac{\partial}{\partial t} a_k(t)$. Dies geschieht mit Hilfe der Heisenbergschen Bewegungsgleichung (2.28) für Vernichter-Operatoren für Elektronen

$$i\hbar \frac{\partial}{\partial t} a_k(t) = \varepsilon_k a_k(t) + \sum_q M_q (b_q(t) + b_{-q}^\dagger(t)) a_{k-q}(t), \quad (3.14)$$

das Einsetzen von (3.14) in (3.13) und einige Umformungen führen dann zu

$$\begin{aligned} i\hbar \frac{\partial}{\partial t} G_k^r(t, t') &= \delta(t - t') + \varepsilon_k G_k^r(t, t') \\ &- \frac{i}{\hbar} \sum_q M_q \langle [(b_q(t) + b_{-q}^\dagger(t)) a_{k-q}(t), a_k^\dagger(t')]_+ \rangle. \end{aligned} \quad (3.15)$$

Der Hamiltonoperator, der die Wechselwirkungen der Elektronen mit den Phononen beschreibt, führt also dazu, dass die Bewegungsgleichungen für die retardierten Greensfunktionen Erwartungswerte höherer Ordnung $\langle [(b_q(t) + b_{-q}^\dagger(t)) a_{k-q}(t), a_k^\dagger(t')]_+ \rangle$ mit Produkten von mehr als zwei Operatoren enthalten. Um diese Terme zu berechnen, werden neben (3.14) die Bewegungsgleichungen für Phononen (2.29) genutzt. In (3.15) sieht man, dass die beiden Ausdrücke

$$\langle b_q(t) [a_{k-q}(t), a_k^\dagger(t')]_+ \rangle \text{ und } \langle b_{-q}^\dagger(t) [a_{k-q}(t), a_k^\dagger(t')]_+ \rangle \quad (3.16)$$

die Erwartungswerte mit Produkten von mehr als zwei Operatoren sind. Für diese lassen sich wieder Bewegungsgleichungen aufstellen:

$$\begin{aligned} i\hbar \frac{\partial}{\partial t} \langle b_q(t) [a_{k-q}(t), a_k^\dagger(t')]_+ \rangle &= (\hbar\omega_{LO} + \varepsilon_{k-q}) \langle b_q(t) [a_{k-q}(t), a_k^\dagger(t')]_+ \rangle \\ &+ M_{-q} \sum_{k'} \langle a_{k'-q}^\dagger(t) a_{k'}(t) [a_{k-q}(t), a_k^\dagger(t')]_+ \rangle \\ &+ \sum_{q'} M_{q'} \langle b_q(t) (b_{q'}(t) + b_{-q'}^\dagger(t)) [a_{k-q-q'}(t), a_k^\dagger(t')]_+ \rangle \end{aligned} \quad (3.17)$$

$$\begin{aligned} i\hbar \frac{\partial}{\partial t} \langle b_{-q}^\dagger(t) [a_{k-q}(t), a_k^\dagger(t')]_+ \rangle &= (-\hbar\omega_{LO} + \varepsilon_{k-q}) \langle b_{-q}^\dagger(t) [a_{k-q}(t), a_k^\dagger(t')]_+ \rangle \\ &+ M_{-q} \sum_{k'} \langle a_{k'-q}^\dagger(t) a_{k'}(t) [a_{k-q}(t), a_k^\dagger(t')]_+ \rangle \\ &+ \sum_{q'} M_{q'} \langle b_{-q}^\dagger(t) (b_{q'}(t) + b_{-q'}^\dagger(t)) [a_{k-q-q'}(t), a_k^\dagger(t')]_+ \rangle \end{aligned} \quad (3.18)$$

Es entstehen also Erwartungswerte mit Produkten von vier Operatoren. Bei einer ähnlichen Rechnung wie der eben beschriebenen würden diese Erwartungswerte an Erwartungswerte noch höherer Ordnung koppeln. An dieser Stelle wird stattdessen die Faktorisierungsnäherung angewendet, das heißt, dass alle auftretenden Erwartungswerte mit Produkten von vier Operatoren in alle möglichen Produkte von Erwartungswerten mit Produkten von zwei Operatoren zerlegt werden. Außerdem werden hier Besetzungsfunktionen für Ladungsträger $f_k(t)$ und Phononen $N_q(t)$ eingeführt:

$$\begin{aligned} \langle a_k^\dagger(t) a_k(t) \rangle &= f_k(t) \\ \langle b_q^\dagger(t) b_q(t) \rangle &= N_q(t) \\ \langle b_q(t) b_q^\dagger(t) \rangle &= 1 + N_q(t) \end{aligned} \quad (3.19)$$

Dabei werden die Annahmen gemacht, dass neben den betrachteten Elektronen keine zusätzlichen Ladungsträger angeregt sind und dass die Phononen als Bad im thermischen Gleichgewicht beschrieben werden können. Es gilt also

$$\begin{aligned} f_k(t) &= 0 \\ N_q(t) \equiv N_{LO} &= \frac{1}{e^{\beta\hbar\omega_{LO}} - 1}. \end{aligned} \quad (3.20)$$

Dabei ist das β von der Temperatur des Systems T abhängig:

$$\beta = \frac{1}{k_B T}, \quad (3.21)$$

wobei k_B die Boltzmann-Konstante ist. Diese Annahmen entsprechen den Eigenschaften, mit denen zu Beginn von Kapitel 2 das Polaron definiert wird. Die Näherungen werden beispielhaft an zwei Erwartungswerten mit Produkten von vier Operatoren demonstriert, die anderen vier werden analog berechnet:

$$\begin{aligned}
\langle a_{k'-q}^\dagger(t) a_{k'}(t) a_{k-q}(t) a_k^\dagger(t') \rangle &\simeq \langle a_{k'-q}^\dagger(t) a_{k'}(t) \rangle \langle a_{k-q}(t) a_k^\dagger(t') \rangle \delta_{q=0} \\
&- \langle a_{k'-q}^\dagger(t) a_{k-q}(t) \rangle \langle a_{k'}(t) a_k^\dagger(t') \rangle \delta_{kk'} \\
&= \underbrace{\langle a_{k'}^\dagger(t) a_{k'}(t) \rangle}_{f_k(t)=0} \langle a_k(t) a_k^\dagger(t') \rangle \\
&- \underbrace{\langle a_{k-q}^\dagger(t) a_{k-q}(t) \rangle}_{f_{k-q}(t)=0} \langle a_k(t) a_k^\dagger(t') \rangle,
\end{aligned} \tag{3.22}$$

$$\langle b_q(t) b_{-q'}^\dagger(t) a_{k-q-q'}(t) a_k^\dagger(t') \rangle \simeq \underbrace{\langle b_q(t) b_{-q'}^\dagger(t) \rangle}_{1+N_q(t)} \langle a_{k-q-q'}(t) a_k^\dagger(t') \rangle \delta_{-q,q'}. \tag{3.23}$$

Letztendlich ergeben sich aus diesen Näherungen die Bewegungsgleichungen

$$\begin{aligned}
i\hbar \frac{\partial}{\partial t} \langle b_q(t) [a_{k-q}(t), a_k^\dagger(t')]_+ \rangle &= (\hbar\omega_{LO} + \varepsilon_{k-q} \langle b_q(t) [a_{k-q}(t), a_k^\dagger(t')]_+ \rangle \\
&+ M_{-q}(1+N) \langle [a_k(t), a_k^\dagger(t')]_+ \rangle,
\end{aligned} \tag{3.24}$$

$$\begin{aligned}
i\hbar \frac{\partial}{\partial t} \langle b_{-q}^\dagger(t) [a_{k-q}(t), a_k^\dagger(t')]_+ \rangle &= (-\hbar\omega_{LO} + \varepsilon_{k-q} \langle b_{-q}^\dagger(t) [a_{k-q}(t), a_k^\dagger(t')]_+ \rangle \\
&+ M_{-q}N \langle [a_k(t), a_k^\dagger(t')]_+ \rangle.
\end{aligned} \tag{3.25}$$

Diese Gleichungen bilden ein Differentialgleichungssystem der Form

$$i\hbar \frac{\partial}{\partial t} K(t) = \varepsilon K(t) + I(t), \tag{3.26}$$

dessen formale Lösung analytisch bestimmt werden kann:

$$K(t) = -\frac{i}{\hbar} \int_{t_0}^t e^{-\frac{i}{\hbar}\varepsilon(t-t')} I(t') dt' + K(t_0). \tag{3.27}$$

Wählt man $t_0 = t'$, liefert $K(t_0)$ keinen Beitrag:

$$K(t') = \langle b_q(t) \underbrace{[a_{k-q}(t'), a_k(t')]}_{=0} \rangle. \tag{3.28}$$

Auf diese Weise erhält man Näherungen für die Erwartungswerte (3.16), die nur noch von Produkten zweier Operatoren abhängen:

$$\begin{aligned}
&\langle b_q(t) [a_{k-q}(t), a_k^\dagger(t')]_+ \rangle \\
&= -\frac{i}{\hbar} M_{-q} \int_{t'}^t e^{-\frac{i}{\hbar}(\hbar\omega_{LO} + \varepsilon_{k-q})(t-t'')} (1+N) \langle [a_k(t''), a_k^\dagger(t')]_+ \rangle dt'',
\end{aligned} \tag{3.29}$$

$$\begin{aligned}
& \langle b_{-q}^\dagger(t) [a_{k-q}(t), a_k^\dagger(t')]_+ \rangle \\
&= -\frac{i}{\hbar} M_{-q} \int_{t'}^t e^{-\frac{i}{\hbar}(-\hbar\omega_{LO} + \varepsilon_{k-q})(t-t'')} (N) \langle [a_k(t''), a_k^\dagger(t')]_+ \rangle dt'' .
\end{aligned} \tag{3.30}$$

Wenn man nun (3.29) und (3.30) in (3.15) einsetzt und M_{-q} mit M_q^* identifiziert, ergeben sich die Bewegungsgleichungen für die retardierten Greensfunktionen für Elektronen in Faktorisierungsnäherung:

$$\begin{aligned}
i\hbar \frac{\partial}{\partial t} G_k^r(t, t') &= \delta(t - t') + \varepsilon_k G_k^r(t, t') - \frac{i}{\hbar} \Theta(t - t') \sum_q M_q M_q^* \left(-\frac{i}{\hbar}\right) \\
&\cdot \int_{t'}^t e^{-\frac{i}{\hbar}(\varepsilon_{k-q})(t-t'')} \langle [a_k(t''), a_k^\dagger(t')]_+ \rangle \\
&\cdot ((1 + N)e^{-i\omega_{LO}(t-t'')} + Ne^{i\omega_{LO}(t-t'')}) dt'' .
\end{aligned} \tag{3.31}$$

Um im Integral die Greensfunktion $G_k^r(t'', t') = -\frac{i}{\hbar} \Theta(t'' - t') \langle [a_k(t''), a_k^\dagger(t')]_+ \rangle$ zu erhalten, wird $\Theta(t - t'') \cdot \Theta(t'' - t')$ eingefügt, was für $t > t'' > t'$ erlaubt ist. Außerdem entspricht der Term $\Theta(t - t'')e^{-\frac{i}{\hbar}(\varepsilon_{k-q})(t-t'')}$ dann der freien retardierten Greensfunktion für Elektronen in einem homogenen System, $G_{k-q}^r(t, t'')|_{\text{free}}$. Das führt zu

$$\begin{aligned}
i\hbar \frac{\partial}{\partial t} G_k^r(t, t') &= \delta(t - t') + \varepsilon_k G_k^r(t, t') + \Theta(t - t') \\
&\cdot \int_{t'}^t \sum_q |M_q|^2 ((1 + N)e^{-i\omega_{LO}(t-t'')} + Ne^{i\omega_{LO}(t-t'')}) \\
&\cdot G_{k-q}^r(t, t'')|_{\text{free}} G_k^r(t'', t') dt'' ,
\end{aligned} \tag{3.32}$$

was der Dyson-Gleichung für retardierte Greensfunktionen

$$(i\hbar \frac{\partial}{\partial t} - \varepsilon_k) G_k^r(t, t') = \delta(t - t') + \int_{t'}^t S_k^r(t, t'') G_k^r(t'', t') dt'' \tag{3.33}$$

mit der Selbstenergie

$$\begin{aligned}
S_k^r(t, t'') &= \\
&= \Theta(t - t'') \sum_q |M_q|^2 ((1 + N)e^{-i\omega_{LO}(t-t'')} + Ne^{i\omega_{LO}(t-t'')}) G_{k-q}^r(t, t'')|_{\text{free}}
\end{aligned} \tag{3.34}$$

entspricht. Dies ist eine nicht selbstkonsistente Random-Phase-Näherung (RPA), da die Selbstenergie $S_k^r(t, t'')$ nur retardierte Greensfunktionen für nicht wechselwirkende Ladungsträger, also die freien retardierten Greensfunktionen für Elektronen enthält. In einer selbstkonsistenten Näherung hängt $S_k^r(t, t'')$ hingegen von den retardierten Greensfunktionen für die wechselwirkenden Ladungsträger ab, an Stelle von $G_{k-q}^r(t, t'')|_{\text{free}}$ steht also $G_{k-q}^r(t, t'')$. Eine solche selbstkonsistente Näherung lässt sich auf anderem Weg herleiten, allerdings übersteigt diese Herleitung den Rahmen der vorliegenden Arbeit. Eine solche Herleitung ist in [10] nachzulesen. An dieser Stelle wird die Ersetzung direkt vorgenommen, um die Wechselwirkung im Rahmen der RPA selbstkonsistent zu beschreiben. Zusätzlich wird hier noch phänomenologisch die Lebensdauer der LO-Phononen eingefügt, indem der Selbstenergie ein Dämpfungsterm der Form

$$e^{-\frac{\Gamma_{LO}}{\hbar}(t-t'')} \tag{3.35}$$

hinzumultipliziert wird, das $\frac{\Gamma_{LO}}{\hbar}$ wurde dabei experimentell bestimmt und beträgt $\frac{0.2}{ps}$ [8]. Somit ergibt sich die Selbstenergie der Dyson-Gleichung in RPA:

$$\begin{aligned} S_k^r(t, t'') &= \\ &= \Theta(t - t'') \sum_q |M_q|^2 \left((1 + N) e^{-i\omega_{LO}(t-t'')} + N e^{i\omega_{LO}(t-t'')} \right) G_{k-q}^r(t, t'') e^{-\frac{\Gamma_{LO}}{\hbar}(t-t'')}. \end{aligned} \quad (3.36)$$

3.3 Umformungen der Dyson-Gleichung

Damit die Dyson-Gleichung besser lösbar wird, werden noch einige weitere Umformungen durchgeführt. Als erstes wird eine Folge des Polaronmodells benutzt. Da das Kristallgitter im Gleichgewicht angenommen wird, spielt es keine Rolle für das System, wann t' und t stattfinden, solange die Differenz $t - t'$ die Gleiche ist. Es lohnt sich also, durch Einführung einer Relativzeit die Dimension der Zeitabhängigkeit zu reduzieren. Dabei bietet sich eine Transformation in Schwerpunkts- und Relativzeit an:

$$\begin{aligned} t_{sp} &= \frac{t+t'}{2} \\ \tau &= t - t', \end{aligned} \quad (3.37)$$

wobei gilt

$$\begin{aligned} \frac{\partial}{\partial t} &= \frac{1}{2} \frac{\partial}{\partial t_{sp}} + \frac{\partial}{\partial \tau} \\ \frac{\partial}{\partial t'} &= \frac{1}{2} \frac{\partial}{\partial t_{sp}} - \frac{\partial}{\partial \tau}. \end{aligned} \quad (3.38)$$

Damit lässt sich die Dyson-Gleichung weiter vereinfachen:

$$(i\hbar \frac{\partial}{\partial \tau} - \varepsilon_k) G_k^r(\tau) = \delta(\tau) + \int_0^\tau \hat{S}_k^r(\tau - \tau') G_k^r(\tau') d\tau'. \quad (3.39)$$

Für die (numerische) Lösung dieses Gleichungssystems wird nicht direkt G_k^r berechnet, sondern die renormierte retardierte Greensfunktion \tilde{G}_k^r , die sich über

$$G_k^r(\tau) = G_{k,0}^r \tilde{G}_k^r(\tau) = -\frac{i}{\hbar} \Theta(\tau) e^{-\frac{i}{\hbar} \varepsilon_k \tau} \tilde{G}_k^r(\tau) \quad (3.40)$$

definiert. Diese Renormierung transformiert die linearen Terme, also die freien Energien ε_k , aus der Gleichung heraus. Dies ist hilfreich für die numerische Lösung, da dadurch triviale Oszillationen der Greensfunktionen entfallen. Damit vereinfacht sich (3.39) zu

$$\frac{\partial}{\partial \tau} \tilde{G}_k^r(\tau) = \int_0^\tau \hat{S}_k^r(\tau - \tau') \tilde{G}_k^r(\tau') e^{-\frac{i}{\hbar} \varepsilon_k (\tau - \tau')} d\tau' \quad (3.41)$$

mit der Selbstenergie

$$\begin{aligned} \hat{S}_k^r(\tau - \tau') &= \sum_q |M_q|^2 \left((1 + N) e^{-i\omega_{LO}(\tau - \tau')} + N e^{i\omega_{LO}(\tau - \tau')} \right) \\ &\cdot e^{-\frac{\Gamma_{LO}}{\hbar}(\tau - \tau')} \tilde{G}_{k-q}^r(\tau - \tau') e^{-\frac{i}{\hbar} \varepsilon_{k-q}(\tau - \tau')}. \end{aligned} \quad (3.42)$$

Da bisher nur der Wetting-Layer als homogenes System betrachtet wurde, wird die Dyson-Gleichung an dieser Stelle auf die Halbleiterstruktur erweitert, die zu Beginn von Kapitel 2 eingeführt wurde. Da für Quantenpunkte keine Implulserhaltung gilt, lässt sich der Zusammenhang $k' = q + k$ nicht anwenden und man muss allgemein über einen weiteren Zustandsindex β summieren:

$$\begin{aligned} \hat{S}_\alpha^r(\tau - \tau') &= \sum_{q,\beta} |M_q^{\alpha\beta}|^2 \left((1 + N_{LO}) e^{-i\omega_{LO}(\tau - \tau')} + N_{LO} e^{i\omega_{LO}(\tau - \tau')} \right) \\ &\cdot e^{-\frac{\Gamma_{LO}}{\hbar}(\tau - \tau')} \tilde{G}_\beta^r(\tau - \tau') e^{-\frac{i}{\hbar}\varepsilon_\beta(\tau - \tau')}. \end{aligned} \quad (3.43)$$

Dabei können α und β sowohl ein Zustand im Wetting-Layer als auch im Quantenpunkt sein. Die Berechnung der Matrixelemente $M_q^{\alpha\beta}$ ist wesentlich anspruchsvoller und wird an dieser Stelle nicht ausgeführt, sondern auf [12] verwiesen.

Diese Gleichung ist eine Integro-Differentialgleichung mit Memory-Kern, wie sie in (1.3) definiert wird, wobei die Selbstenergie multipliziert mit der Transformation (3.40) $\hat{S}_k^r(\tau - \tau') \cdot e^{-\frac{i}{\hbar}\varepsilon_k(\tau - \tau')}$ dem $B(t - t', y(t - t'))$ und die retardierten Greensfunktionen dem $y(t)$ entsprechen. Der Startwert des zugehörigen Anfangswertproblems ist

$$\tilde{G}_k^r(0) = 1, \quad (3.44)$$

was sich aus den Kommutatorrelationen für die retardierten Greensfunktionen herleiten lässt. Im weiteren Verlauf der Arbeit wird ausschließlich von den Bezeichnungen $B(t - t', y(t - t'))$ und $y(t)$ Gebrauch gemacht, um die Betrachtungen auf die mathematischen Aspekte der Integro-Differentialgleichungen zu fokussieren.

4 Andere Lösungsansätze

Es sind verschiedene Lösungsansätze denkbar, um Integro-Differentialgleichungen mit Memory-Kern numerisch zu behandeln. Bei der numerischen Lösung von Integro-Differentialgleichungen sind zwei Grundaufgaben zu verrichten:

1. Lösung der Differentialgleichung und
2. Berechnung des Memoryintegrals

In diesem Kapitel werden einige Klassen von Verfahren diskutiert und es wird erläutert, warum diese Lösungsansätze schlechter geeignet sind als die Variante des ABM113, die in Kapitel sechs und sieben beschrieben wird.

- Solver mit konstanten Schrittweiten für Differentialgleichung und Integral

Die einfachsten Verfahren, um Integro-Differentialgleichungen mit Memory-Kern zu behandeln, sind solche mit konstanten Schrittweiten für beide Aufgaben. Hierbei wird die Differentialgleichung auf einem festen Gitter gelöst, der Funktionswert jedes Schrittes gespeichert und das Integral anschließend auf dem selben Gitter gelöst. Dieses Verfahren hat den Vorteil, dass alle Punkte, die für die Integration benötigt werden, auf jeden Fall vorhanden sind. Außerdem ist es besonders einfach, einen Vergangenheitsspeicher für den Memory-Kern anzulegen, da das Integrationsgitter bekannt ist. Mit einem Vergangenheitsspeicher für den Memory-Kern kann die Integration für kompliziertere Kerne signifikant beschleunigt werden, da nicht in jedem Schritt auf dem gesamten Gitter die Kernausswertung wiederholt werden muss.

Ein Beispiel für diese Art Verfahren ist es, für die Differentialgleichung ein Prädiktor-Korrektor-Verfahren mit konstanter Schrittweite [2] und für das Integral eine zusammengesetzte Trapezregel [2] zu verwenden. Für ausreichend kleine Schrittweiten lassen sich mit dieser Kombination die meisten Probleme lösen, allerdings wächst der Zeitaufwand quadratisch mit der Anzahl der Punkte des zugrunde gelegten Gitters, vergleiche hierzu auch Kapitel 1.2. Die Lösung einfacher Probleme oder solcher, bei denen die benötigte Zeit nicht problematisch ist, ist mit Verfahren mit konstanten Schrittweiten sinnvoll möglich, allerdings ist der Zeitaufwand für kompliziertere Memory-Kerne, die ein sehr feines Integrationsgitter benötigen, sehr hoch, da das Differentiationsgitter genauso fein sein muss und somit die Anzahl der Auswertungen der rechten Seite der Differentialgleichung stark steigt. Weil bei jeder dieser Auswertungen das Integral gelöst werden muss, benötigt das Verfahren dann möglicherweise sehr viel Zeit.

Runge-Kutta-Verfahren sind für die Lösung dieser Problemstellung nicht geeignet, da in jedem Schritt eines Runge-Kutta-Verfahrens zusätzlich Funktionsauswertungen halber Schrittweiten benötigt werden. Diese Funktionsauswertungen müssen jeweils auf dem Gitter interpoliert werden, wodurch zusätzliche Fehler und unnötiger zeitlicher Mehraufwand entstehen.

- Adaptive Differentialgleichungs-Solver und konstante Integrationsgitter

Vor allem bei Problemen, deren Änderungen der Funktionswerte im Verlaufe der Zeit geringer werden, bietet es sich an, einen Solver mit adaptiver Schrittweite zu benutzen. Durch die Adaptation der Schrittweite werden Bereiche der Funktion, in denen sich der Funktionswert langsamer ändert, geringer aufgelöst. Dadurch können gegenüber einem konstanten Verfahren Differentialauswertungen eingespart werden, die, wie in Kapitel 1.2 erläutert, den Großteil der Zeit erfordern. Ein weiterer Vorteil von adaptiven Solvern ist, dass man eine (lokale) Genauigkeit fordern kann.

Die Problematik von adaptiven Solvern liegt darin, dass es kein vorhersehbares Gitter gibt, auf dem Funktionsauswertungen stattfinden und somit keine Werte für $y(t')$ vorhanden sind, um $B(t - t', y(t - t'))$ zu berechnen. Dadurch ist es notwendig, für die Integration Funktionswerte geeignet interpolieren zu können. Für diese Aufgabe ist das Verfahren von Adams-Bashforth-Moulton in der Variante von Shampine und Gordon [11] geeignet, welches ein Prädiktor-Korrektor-Verfahren mit adaptiver Schrittweite und Ordnung der verwendeten Polynome ist. Dieser Solver liefert die Interpolationspolynome, die für die Integration benötigt werden, natürlicherweise mit. Eine Möglichkeit ist es, diesen Solver zu verwenden, nach jedem erfolgreichen Schritt Funktionswerte y_n auf einem konstanten Gitter zu interpolieren, zu speichern und auf diesem Gitter eine zusammengesetzte Trapezregel zu verwenden. Dadurch wird der Vorteil des Geschwindigkeitsgewinns der Adaptation durch möglichst wenige Funktionsauswertungen mit den Vorteilen eines konstanten Gitters in der Integration verbunden, allerdings findet die Integration immer noch auf einem möglicherweise sehr feinen, konstanten Gitter statt, was viel Zeit benötigt. Dieses Verfahren wurde am ITP-Bremen realisiert und wird in Kapitel 7 mit den Ergebnissen des in dieser Arbeit konstruierten Solvers verglichen.

- Adaptive Integral-Solver

Eine naheliegende Verbesserungsmöglichkeit zu konstanten Integrationsgittern ist es, adaptive Schrittweitensteuerung für das Integral zuzulassen und somit die Berechnung zu beschleunigen. Ein Beispiel für ein solches Verfahren ist das Gauß-Kronrod-Verfahren [13], bei dem ein Gauß-Verfahren mit sieben Auswertungen des Integranden in ein Gauß-Verfahren mit 15 Auswertungen eingebettet wird und diese beiden Verfahren zur Fehlerabschätzung verglichen werden. Allerdings sind solche Verfahren ungeeignet für die Berechnung von Integralen im Verlaufe der Lösung von Integro-Differentialgleichungen, da nicht bekannt ist, auf welchem Gitter die nächsten Auswertungen stattfinden werden. Deshalb müssen in jedem Schritt der Differentialgleichung alle Punkte, die für die Integration benötigt werden, neu interpoliert werden, wodurch der Aufwand für kompliziertere Probleme größer ist als bei einer zusammengesetzten Trapezregel auf einem konstantem, gleich bleibenden Integrationsgitter. Hinzu kommt, dass beim Einsatz von zwei adaptiven Solvern für Differentialgleichung und Integral das Problem entsteht, dass der eine Solver versucht, die numerischen Fehler des anderen aufzulösen. Diese Problematik lässt sich durch Feinjustierung der geforderten Genauigkeiten reduzieren, allerdings muss diese Justierung für jede Integro-Differentialgleichung spezifisch durchgeführt werden, was eine benutzerfreundliche Implementierung unmöglich macht.

5 Adams - Bashforth - Moulton

5.1 Allgemeine Idee

Das Prädiktor-Korrektor-Verfahren nach Adams-Bashforth-Moulton, das in dieser Arbeit als Grundlage für die Lösung der Integro-Differentialgleichung genutzt wird, ist ein adaptives Mehrschrittverfahren, das die Schrittweite in jedem Schritt verdoppeln oder halbieren und die Ordnung der verwendeten Interpolationspolynome von eins bis dreizehn variieren kann. Die verwendete Implementierung richtet sich nach der von Shampine und Gordon [11] in FORTRAN77 in einer objektorientierten, allgemeine Typen und Algebren unterstützenden Überarbeitung des Autors dieser Arbeit in FORTRAN2003. Dieser Solver wird im Folgenden mit ABM113 bezeichnet.

Das Ziel des ABM113 ist es, durch Variation der Schrittweite und der Ordnung der Interpolationspolynome die Anzahl der Funktionsauswertungen zu minimieren, was für Integro-Differentialgleichungen ideal ist. Der entstehende Overhead ist, anders als im Jahr 1976, als der Solver erstmals veröffentlicht wurde, angesichts aktueller Rechner zu vernachlässigen.

Im folgenden Kapitel werden einige grundlegende Überlegungen des ABM113 dargelegt, die größtenteils aus [11] stammen, und anschließend, im nächsten Kapitel, die erforderlichen Modifikationen zur Lösung von Integro-Differentialgleichung ausgeführt.

5.2 Polynomiale Interpolation

Für den ABM113 ist es notwendig, vorhandene Interpolationspolynome $P_k(x)$ der Ordnung k für eine Funktion $f(x)$ über paarweise verschiedene Stützstellen x_1, x_2, \dots, x_k um einen weiteren, zu allen bisherigen paarweise verschiedenen Punkt x_{k+1} mit $P_{k+1}(x_{k+1}) = f(x_{k+1})$ zu erweitern. Ziel ist also eine Form

$$P_{k+1}(x) = P_k(x) + R_{k+1}(x), \quad (5.1)$$

wobei $R_{k+1}(x)$ ein Polynom von der Ordnung $k+1$ ist und $f(x)$ von $P_{k+1}(x)$ auf den Stützstellen $x_1, x_2, \dots, x_k, x_{k+1}$ interpoliert wird. Ein Interpolationspolynom von Ordnung k ist auf k Stützstellen eindeutig definiert, eine solche Konstruktion ist also zulässig und algebraisch äquivalent mit allen anderen möglichen Darstellungen. Konkret wird diese Darstellung erreicht, indem für die Polynome der ersten und zweiten Ordnung

$$\begin{aligned} P_1(x) &= f_1 \\ P_2(x) &= P_1(x) + R_2(x) \end{aligned} \quad (5.2)$$

gebildet wird, wobei

$$R_2(x) = (x - x_1)a_1 \quad (5.3)$$

gilt, wodurch automatisch $P_2(x_1) = f_1$ erfüllt ist. Außerdem soll $f(x_2) = f_2 = P_2(x_2)$ gelten, das führt zu

$$f_2 = f_1 + (x_2 - x_1)a_1 \Rightarrow a_1 = \frac{f_2 - f_1}{x_2 - x_1}. \quad (5.4)$$

Daraus ergibt sich induktiv, dass für den neuen Teil $R_{k+1}(x)$ des Polynoms $P_{k+1}(x)$

$$R_{k+1}(x) = (x - x_1)(x - x_2) \dots (x - x_k) a_k \text{ mit } a_k = \frac{f_{k+1} - P_k(x_k)}{\prod_{j=1}^k (x_{k+1} - x_j)} \quad (5.5)$$

gilt. Außerdem wird die Bezeichnung $f[x_1, \dots, x_k]$ für a_k eingeführt, die deutlich macht, von welchen x_i es abhängt. In dieser Notation haben die Interpolationspolynome dann die Form

$$P_k(x) = f[x_1] + (x - x_1)f[x_1, x_2] + \dots + (x - x_1)(x - x_2) \dots (x - x_{k-1})f[x_1, \dots, x_k]. \quad (5.6)$$

Diese nennt man die dividierte Differenzen-Darstellung oder auch Newton-Darstellung von Polynomen. Ein weiterer Vorteil dieser Darstellung ist, dass die f für die nächste Ordnung sehr einfach gebildet werden können, es gilt:

$$f[x_1, \dots, x_k] = \frac{f[x_1, \dots, x_{k-1}] - f[x_2, \dots, x_k]}{x_1 - x_k}. \quad (5.7)$$

Es sei außerdem angemerkt, dass die Reihenfolge, in der die Stützstellen hinzugefügt werden, für das resultierende Polynom keine Auswirkungen haben, es ist äquivalent möglich, mit $f[x_k]$ als konstantem Term zu arbeiten. Die Genauigkeit eines solchen Interpolationspolynoms auf einem Intervall $[a, b]$ beschreibt der folgende Satz:

Satz

Sei $f(x) \in C^k[a, b]$, $x_1, \dots, x_k \in [a, b]$ paarweise Verschieden, f_1, \dots, f_k die zugehörigen Funktionswerte von $f(x)$ und $P_k(x)$ das Polynom von Ordnung $k - 1$, das diese Punktepaare interpoliert.

Sei außerdem $\omega(x) = \prod_{i=1}^k (x - x_i)$. Dann gilt für alle $x \in [a, b]$:

$$f(x) - P_k(x) = E(x)\omega(x) \quad (5.8)$$

wenn für $E(x)$

$$E(x) = \begin{cases} \frac{f'(x_i) - P'_k(x_i)}{\omega'(x_i)} & \text{für } x = x_i \\ \frac{f^{(k)}(\xi(x))}{k!} & \text{für } x \neq x_i \end{cases} \quad (5.9)$$

gilt und $\xi(x)$ ein unbekannter Punkt aus dem kleinsten Intervall ist, das x und alle Datenpunkte x_i für $i = 1, \dots, k$ enthält.

Der Beweis dieses Satzes verwendet die Regel von L'Hospital und den Satz von Rolle und ist in [11] nachzulesen.

5.3 ABM113

Der ABM113 nutzt ein PECE-Verfahren, das eine Kombination eines expliziten Verfahrens (dem Prädiktor) und eines impliziten Verfahrens (dem Korrektor) ist. PECE steht für Predict-Evaluate-Correct-Evaluate. Dabei werden folgende Schritte vom Solver für jeden Zeitschritt durchgeführt:

- Prädiktor-Schritt

Im Prädiktor-Schritt wird eine Schätzung p_{n+1} von y_{n+1} erzeugt, indem

$$p_{n+1} = y_n + \int_{x_n}^{x_{n+1}} P_{k,n}(t) dt \quad (5.10)$$

berechnet wird. Dieses explizite Verfahren wird Adams-Bashforth-Verfahren genannt und erzeugt aus der Interpolation der *Ableitung* der gesuchten Funktion für die letzten k Punkte, $P_{k,n}(x)$, (der zusätzliche Index n entspricht dem aktuellen Schrittindex) eine Schätzung für y_{n+1} für den nächsten Punkt x_{n+1} .

- Erster Funktionsauswertungsschritt

Nach der Berechnung von p_{n+1} wird im ersten Funktionsauswertungsschritt die rechte Seite der Differentialgleichung $w_{n+1}^p = w(x_{n+1}, p_{n+1})$ ausgewertet und das Polynom $P_{k,n}(x)$ mit f_{n+1}^p zu $P_{k+1,n}^*(x)$ aktualisiert. Für das Interpolationspolynom $P_{k+1,n}^*(x)$ gilt dann

$$\begin{aligned} P_{k+1,n}^*(x_{n+1-j}) &= w_{n+1-j} \text{ für } j = 1, \dots, k \\ P_{k+1,n}^*(x_{n+1}) &= w_{n+1}^p. \end{aligned} \quad (5.11)$$

- Korrektor-Schritt

Anschließend wird im Korrektor-Schritt die Schätzung für y_{n+1} verbessert, indem

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} P_{k+1,n}^*(t) dt \quad (5.12)$$

gebildet wird. Dieses implizite Verfahren wird Adams-Moulton-Methode genannt.

- Zweiter Funktionsauswertungsschritt

Wie im ersten Evaluate-Schritt wird im zweiten Funktionsauswertungsschritt die rechte Seite der Differentialgleichung $w_{n+1} = w(x_{n+1}, y_{n+1})$ (vgl (1.3)) ausgewertet und das Polynom $P_{k,n}(x)$ mit w_{n+1} zu $P_{k+1,n+1}(x)$ aktualisiert. Für das Interpolationspolynom $P_{k+1,n+1}(x)$ gilt analog

$$P_{k+1,n+1}(x_{n+2-j}) = w_{n+2-j} \text{ für } j = 1, \dots, k. \quad (5.13)$$

Dieses Verfahren kann so mit Polynomen beliebiger Ordnung durchgeführt werden, allerdings zeigt sich, dass Polynome für den Prädiktor-Schritt mit mehr als zwölfter Ordnung Probleme mit der Stabilität des Solvers verursachen [11]. Als Korrektor nutzt der ABM113 grundsätzlich ein Verfahren, das eine Ordnung höher ist als der im selben Zeitschritt verwendete Prädiktor. Dadurch ergibt sich auch der Name des Solvers, 113, der darauf hinweist, dass der Solver zwischen Verfahren erster und dreizehnter Ordnung variieren kann. Die dreizehnte Ordnung wird allerdings nur im Korrektor Schritt erreicht.

An dieser Stelle sei als veranschaulichendes Beispiel für ein PECE-Verfahren ein Prädiktor mit Polynomordnung eins und ein Korrektor mit Polynomordnung zwei ausgeführt:

$$\begin{aligned}
 p_{n+1} &= y + h_{n+1}w_n \\
 w_{n+1}^p &= w(x_{n+1}, p_{n+1}) \\
 y_{n+1} &= y_n + \frac{h_{n+1}}{2}(w_{n+1}^p + w_n) \\
 w_{n+1} &= w(x_{n+1}, y_{n+1}).
 \end{aligned} \tag{5.14}$$

Der Prädiktor mit Ordnung eins entspricht einem Euler-Verfahren, der Korrektor der Ordnung zwei einer Trapezregel. Im Code bietet es sich an, nicht x_n, x_{n-1}, \dots als grundlegende Größen zu benutzen, sondern, mit Blick darauf, dass sie variieren sollen, Schrittweiten $h_i = x_i - x_{i-1}$ und Summen von Schrittweiten $\Psi_i(n+1) = \sum_{j=n+1-i}^n h_j$. Außerdem wird die Größe $\Phi_i(n)$ eingeführt:

$$\begin{aligned}
 \Phi_1(n) &= w_n \\
 \Phi_i(n) &= \Psi_1(n)\Psi_2(n)\dots\Psi_{i-1}(n)w[x_n, x_{n-1}, \dots, x_{n-i+1}] \text{ für } i > 1.
 \end{aligned} \tag{5.15}$$

Diese Größen werden im ABM113 nach jedem erfolgreichen Schritt bereitgestellt. In den folgenden Kapiteln wird das Argument n aus Gründen der Übersicht oft weggelassen, es ist dann immer das Φ und das Ψ des aktuellen Schrittes der Differentialgleichung gemeint.

6 Modifikationen des ABM113

Im Verlaufe der Entwicklung des ABM113 werden Interpolationspolynome aufgebaut, die die gesuchte Funktion mit der geforderten Genauigkeit approximieren. Dass dies möglich ist, zeigt der Satz in Kapitel (5.2), der eine Abschätzung des Interpolationsfehlers liefert.

Die Idee dieser Arbeit ist es, diese Polynome für die Integration zu benutzen, da die Multiplikation und Integration von Polynomen analytisch sehr einfach ist. Dadurch, dass die geforderte Genauigkeit für das Interpolationspolynom auf dem gesamten Intervall gilt entstehen durch die Integration keine zusätzlichen Fehler oberhalb der Maschinengenauigkeit. Die Genauigkeit der Lösung der Integro-Differentialgleichung entspricht damit der für den ABM113 geforderten Genauigkeit, solange diese signifikant größer ist als die Maschinengenauigkeit, das heißt um etwa zwei Größenordnungen. Der ABM113 testet dies und gibt eine Warnung aus, wenn Genauigkeiten lokal in diesen Bereich an die Maschinengenauigkeit herankommen. Diese Methode hat außerdem den Vorteil adaptiv zu sein, da die Quadratur stückweise auf kleinsten Schnittmengen der stückweise definierten Polynome für $y(t)$ und $B(t - t', y(t - t'))$ stattfindet. Dadurch wird die Berechnung der Integrale gegenüber einer Integration auf einem konstanten Gitter möglicherweise beschleunigt.

Um den ABM113 für Integro-Differentialgleichungen mit Memory-Kern zu optimieren, müssen einige Modifikationen durchgeführt werden:

- Aufbau geeigneter Interpolationspolynome für die Vergangenheit von $y(t')$ und $B(t - t', y(t - t'))$
- Speichern der Interpolationspolynome
- Realisierung der Multiplikation und Integration von $y(t')$ und $B(t - t', y(t - t'))$

Diese Modifikationen werden in diesem Kapitel genauer erläutert.

6.1 Aufbau der Interpolationspolynome

- Zur Verfügung stellen der Interpolationspolynome für $y(t')$ während der Funktionsauswertungsschritte

Während der Propagation des ABM113 werden Polynome der Ordnung k aufgebaut, die $\frac{d}{dt}y(t)$ auf den letzten k Schritten interpolieren. Diese Polynome werden am *Ende* eines erfolgreichen Schrittes in einer Form zur Verfügung gestellt, die eine elegante Interpolation von $y(t)$ auf den letzten k Schritten und damit auf dem Intervall $[x_{n+1-k}, x_n]$ erlaubt. Der ABM113 muss aber in jedem Schritt zweimal das Differential und somit das Integral für die Vergangenheit auswerten, insbesondere auch für das neueste Intervall $[x_n, x_{n+1}]$. Für dieses Intervall stellt der ABM113 in seiner ursprünglichen Form während der Funktionsauswertungen im Verlaufe eines Schrittes *keine* Interpolationsmöglichkeiten für $y(t)$ zur Verfügung, da die Interpolationspolynome für $\frac{d}{dt}y(t)$ während des Prädiktor- und Korrektors-Schrittes speziell umgebaut werden, um die Schätzungen zu vereinfachen. Aus diesem Grund werden zusätzlich zu den Dividierten

Differenzen für $\frac{d}{dt}y(t)$ auch Dividierte Differenzen für $y(t)$ aufgebaut, die vor jeder Funktionsauswertung aktualisiert werden. Das geschieht während des PECE Verfahrens (vgl. Kapitel 5.3) auf folgende Weise:

1. Vor dem Prädiktor-Schritt wird das Interpolationspolynom $P_{k,n}^y(x)$ für $y(t)$ zusätzlich gesichert.
2. Im Prädiktor-Schritt werden die Dividierten Differenzen $P_{k,n}^y(x)$ mit der Schätzung des Prädiktors für y_{n+1} temporär aktualisiert.
3. Nach dem ersten Funktionsauswertungsschritt wird der gesicherte Zustand des Polynoms von vor dem Prädiktor-Schritt wieder hergestellt.
4. Im Korrektor-Schritt werden die Dividierten Differenzen $P_{k,n}^y(x)$ mit der Schätzung des Korrektors für y_{n+1} aktualisiert und somit $P_{k+1,n+1}^y(x)$ gebildet.

Das Hinzufügen der Schätzung y_{n+1} des Prädiktor- bzw Korrektor-Schrittes zu den Dividierten Differenzen erfolgt dabei wie folgt (vergleiche für die Bezeichnungen (5.15) und für die Entwicklungsformel (5.7)):

$$\begin{aligned}
 \Phi_1^y(n+1) &= y_{n+1} \\
 \Phi_i^y(n+1) &= \Psi_1(n+1) \cdot \dots \cdot \Psi_i(n+1) f[x_{n+1}, \dots, x_{n-i+1}] \\
 &= \Psi_1(n+1) \cdot \dots \cdot \Psi_i(n+1) \frac{f[x_{n+1}, \dots, x_{n-i+2}] - f[x_n, \dots, x_{n-i+1}]}{\Psi_i(n+1)} \\
 &= \Phi_i(n+1) - \frac{\Psi_1(n+1) \cdot \Psi_2(n+1) \cdot \dots \cdot \Psi_{i-1}(n+1)}{\Psi_1(n) \cdot \dots \cdot \Psi_{i-1}(n)} \Phi_i(n).
 \end{aligned} \tag{6.1}$$

Die Ordnung und die Ψ 's des Interpolationspolynoms für $y(t)$ werden von dem Interpolationspolynom für $\frac{d}{dt}y(t)$ übernommen. Auf diese Weise wird gewährleistet, dass auch für den neuesten Schritt bei jeder Funktionsauswertung Interpolationspolynome für $y(t)$ in geeigneter Form zur Verfügung stehen.

- Interpolationspolynome für B

Die Auswertung der Selbstenergie, die im physikalischen Modell hinter dem $B(t-t', y(t-t'))$ steht, ist relativ aufwändig, zumindest gemessen an der Menge der benötigten Funktionsauswertungen. Deshalb ist es hilfreich, das $B(t-t', y(t-t'))$ geeignet zu interpolieren. Weil diese Interpolationspolynome für $B(t-t', y(t-t'))$ später mit den Interpolationspolynomen für $y(t)$ multipliziert werden sollen, bietet es sich an, für $B(t-t', y(t-t'))$ eine zusätzliche Differentialgleichung

$$\frac{d}{d(t-t')} X(t-t') = B(t-t', y(t-t')) \tag{6.2}$$

mit dem ABM113 zu lösen, um Polynome der gleichen Form wie für $y(t)$ zu erhalten. Die Lösung dieser Differentialgleichung lautet

$$X(t-t') = \int_0^{t-t'} B(s, y(s)) ds, \tag{6.3}$$

und im ABM113 werden die Interpolationspolynome für $\frac{d}{d(t-t')}X(t-t')$ (vgl. Kapitel 5.3) aufgebaut

$$\frac{d}{d(t-t')}X(t-t') = \frac{d}{d(t-t')} \int_0^{t-t'} B(s, y(s)) ds = B(t-t', y(t-t')) \quad (6.4)$$

und somit eine Interpolation für $B(t-t', y(t-t'))$ von der gleichen Form und Genauigkeit (wenn gewünscht) wie für die eigentlich zu lösende Differentialgleichung durchgeführt. Für die Berechnung von $B(t-t', y(t-t'))$ wird $y(t-t')$ benötigt, deshalb erfolgt die Berechnung der Differentialgleichung (6.2) direkt nach jeder Speicherung des Interpolationspolynoms für $y(t)$. Dabei ist zu beachten, dass die Speicherung von $B(t-t', y(t-t'))$ eines Schrittes nur dann erfolgt, wenn der vom ABM113, der die Differentialgleichung für $B(t-t', y(t-t'))$ löst, geschätzte nächste Zeitpunkt erreicht wird und das Polynom, das für $y(t)$ gebaut wird, ein finales, das heißt eines aus dem Korrektor-Schritt ist. Andernfalls wird immer der zuletzt gespeicherte Schritt des ABM113 geladen, in dem das letzte Mal ein solcher Fall eingetreten ist und die Propagation von diesem Punkt bis zum aktuellen x_{n+1} erneut gestartet. Dadurch wird das Gitter der Differentialgleichung für $B(t-t', y(t-t'))$ unabhängig vom Gitter der Differentialgleichung für $y(t)$ aufgebaut. Dieses Vorgehen ist hilfreich, da die stärkste Änderung von $y(t)$ zu Beginn und die stärkste Änderung von $B(t-t', y(t-t'))$ am Ende der Integration stattfindet, dies wird in Kapitel 7.3 genauer erläutert. Ein gemeinsames Gitter für die Interpolationen wäre somit am Anfang und am Ende der Integration zu genau.

Die unterschiedlichen Gitter können allerdings dazu führen, dass die Stellen der Funktionsauswertung von $B(t-t', y(t-t'))$ sehr nah an Punkten des Vergangenheitspeichers von $y(t)$ liegen. Das hat zur Folge, dass der ABM113, der die Interpolationspolynome für das $B(t-t', y(t-t'))$ aufbaut, mit einer sehr kleinen Schrittweite ausgeführt wird. Wenn die Schrittweite kleiner ist als vier mal die Maschinengenauigkeit multipliziert mit dem Betrag der aktuellen Zeit

$$h < 4 * eps * |t|, \quad (6.5)$$

kann der ABM113 nicht ausgeführt werden. Deshalb wird in solchen Fällen kein Polynom erzeugt und bei der Multiplikation und Integration auf dem entsprechenden Intervall gleich Null gesetzt. Da diese Intervalle eine Breite in der Größenordnung 10^{-15} besitzen und $y(t)$ (physikalisch) durch eins nach oben beschränkt ist, ist das Integral über diesem Intervall ebenfalls sehr klein und spielt bei der Berechnung des Integrals über alle Intervalle keine Rolle.

6.2 Organisation des Vergangenheitspeichers

- Speicherbedarf reduzieren

Die Speicherung von $y(t)$ und $B(t-t', y(t-t'))$ erfolgt so lange an derselben Stelle im Vergangenheitspeicher, bis die Anzahl der verwendeten x_n seit der letzten neuen Stelle der Ordnung des Polynoms entspricht und somit der interpolierte Bereich sich nicht mehr mit dem an der vorherigen Stelle gespeicherten überschneidet. Dadurch wird der Speicherbedarf, je nach Ordnung der Interpolationspolynome, auf bis zu einem Zwölftel reduziert.

- Finden des richtigen Speicherindex

Um den konkreten Index n^* zu einem t^* mit $x_{n^*} \leq t^* \leq x_{n^*+1}$, bei dem das zugehörige Interpolationspolynom gespeichert ist, zu finden, wird ein Vektor angelegt, der mit einem Skalierungsfaktor s_f wie folgt gebildet wird:

$$v(\lfloor x_{n^*} s_f \rfloor : \lfloor x_{n^*+1} s_f \rfloor) = n^* + 1, \quad (6.6)$$

so dass für ein gegebenes t^* der entsprechende Index n^* schnell gefunden werden kann, indem $v(\lfloor t^* s_f \rfloor)$ gebildet wird. Der Skalierungsfaktor beeinflusst dabei, wie genau die Zeit aufgelöst wird und wie stark anschließend nachjustiert werden muss, um den korrekten Index zu finden. Wenn die Schätzung zu ungenau ist und zuviel nachjustiert werden muss, empfiehlt es sich, den Skalierungsfaktor und somit die Auflösung zu erhöhen, da es durch die Nachjustierung leicht zu Geschwindigkeitsverlusten kommen kann. Dies ist natürlich nur dann sinnvoll, wenn es dadurch nicht zu Speicherproblemen kommt. Ein geeigneter Wert für das vorliegende physikalische Problem ist $s_f = 100000$.

- Auswahl des richtigen $P_{k,n}^B(t - t')$ bei stückweiser Multiplikation mit einem $P_{k,n}^y(t')$

Bei der Multiplikation der Polynome für $y(t')$ und $B(t - t', y(t - t'))$ müssen zwei Dinge beachtet werden: Das Polynom $P_{k',n'}^B(\tau)$ für $B(t - t', y(t - t'))$ ist über $\tau = t - t'$ gespeichert, da es beim Ausführen des ABM113 nach und nach für wachsende τ aufgebaut wird. Dieses $\tau = t - t'$ entspricht für ein fixiertes t^* im Bezug auf t' einer Verschiebung um t^* nach links und einer Spiegelung um die y -Achse, also wird der Graph auf dem Intervall $[0, t^*]$ umgedreht. In Abbildung 2 wird dies am Beispiel von $t^* = 10$ und $B(\tau, y(\tau)) = -\tau$ verdeutlicht, die analytische Lösung dieses Problems wird in Kapitel 7.2 ausführlicher behandelt.

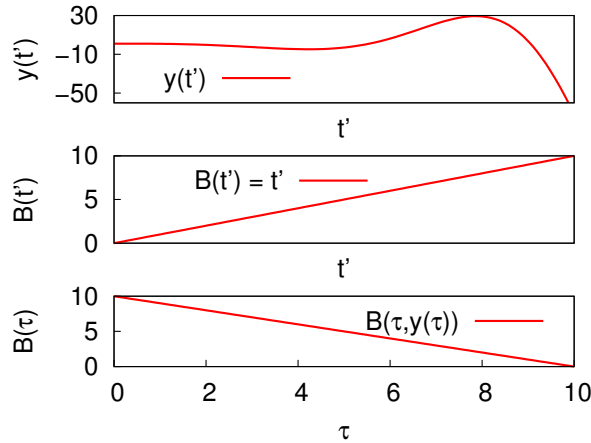


Abbildung 2: Im ersten Plot ist die Lösung der Integro-Differentialgleichung für $B(\tau, y(\tau)) = -\tau$, im zweiten der Memory-Kern für fixiertes $t^* = 10$ in Abhängigkeit von t' und im dritten $B(\tau, y(\tau)) = -\tau$ dargestellt. Der ABM113 speichert die Interpolationspolynome für den oberen und unteren Graphen, multipliziert werden aber Interpolationspolynome für den oberen und der mittleren Graphen.

Außerdem sind die Gitter zwischen 0 und t für $B(t - t', y(t - t'))$ und $y(t')$ nicht identisch. Deshalb können jeweils nur Polynome auf kleinsten Schnittmengen der Intervalle der Gitter multipliziert und quadriert werden. Die Auswahl der Intervalle erfolgt dabei Stück für Stück immer erst für $P_{k,n}^y(t')$. Anschließend werden mit Hilfe des Vektors (6.6) die Indices der Intervalle $P_{k',m}^B(t - t')$ gesucht die das Intervall von $P_{k,n}^y(t')$ überdecken. Dabei ist zu beachten, dass die Polynome immer im oberen Index bezüglich der jeweiligen Zeitrechnung gespeichert sind, dies wird in Abbildung 3 und 4 durch die Richtung der gestrichelten Pfeile verdeutlicht. Bei der Auswahl der Intervalle für $P_{k',m}^B(t - t')$ sind zwei Fälle zu betrachten:

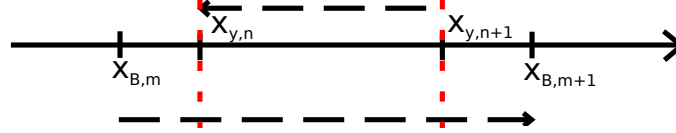


Abbildung 3: Stellt ist den Fall dar, dass ein Intervall von $P_{k',m}^B(t - t')$ existiert, dass das gewählte Intervall von $P_{k,n}^y(t')$ vollständig überdeckt.

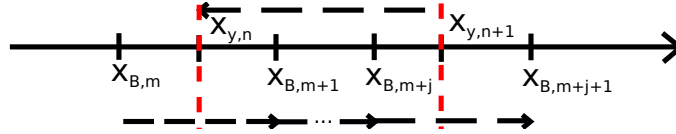


Abbildung 4: Zeigt den Fall, dass Grenzen von Intervallen von $P_{k',m}^B(t - t')$ innerhalb des Intervalls von $P_{k,n}^y(t')$ liegen.

Die Multiplikation von $P_{k,n}^y(t')$ und $P_{k',n'}^B(t - t')$ erfolgt dann stückweise auf Intervallen

$$[x_{y,n}, x_{B,m+1}], [x_{B,m+1}, x_{B,m+2}], \dots, [x_{B,m+j-1}, x_{B,m+j}], [x_{B,m+j}, x_{y,n+1}] \quad (6.7)$$

wobei für j gilt:

$$x_{y,n} \leq x_{B,m+1} \leq x_{B,m+j} \leq x_{y,n+1}. \quad (6.8)$$

6.3 Multiplikation und Integration, Variante eins

Ein erster natürlicher Ansatz ist, die Polynome für $y(t')$ und $B(t - t', y(t - t'))$ so umzubauen, dass Integration und Multiplikation analytisch möglichst einfach wird.

- Umbauen der Interpolationspolynome in eine einfachere Form

Die Polynome für $B(t - t', y(t - t'))$ und $y(t)$ werden ursprünglich in der Form

$$P_k(x) = \sum_{i=1}^k \Phi_i \prod_{m=1}^{i-1} \frac{x - x_n + \Psi_{m-1}}{\Psi_m} \quad (6.9)$$

aufgebaut, wobei k die Ordnung des Polynoms ist und $\Psi_0 = 0$ angenommen wird. Die Definitionen von Φ und Ψ werden in (5.15) näher erläutert. Mit den Definitionen

$$d_i = \frac{\Phi(i)}{\prod_{l=1}^{i-1} \Psi_l} \text{ und } c_m = \begin{cases} -x_n & \text{für } m = 1 \\ -x_n + \Psi_{m-1} & \text{für } m > 1 \end{cases} \quad (6.10)$$

gilt für die Interpolationspolynome für $B(t - t', y(t - t'))$ und $y(t')$:

$$P_{k,n}(x) = \sum_{i=1}^k d_i \prod_{m=1}^{i-1} (x + c_m). \quad (6.11)$$

Die Polynome (6.11) werden anschließend mit einem inversen Horner Schema in die Form

$$P_k(x) = \sum_{i=1}^{k+1} g_i x^{i-1}, \quad (6.12)$$

gebracht. Am Ende jedes erfolgreichen Schrittes des ABM113 werden dann die Koeffizienten g_i und die Ordnung des Polynoms k gespeichert.

- Multiplikation und Integration

Bei der Multiplikation der Interpolationspolynome muss beachtet werden, dass nicht $P_{k_B}^B(t')$, sondern $P_{k_B}^B(t - t')$ mit $P_{k_y}^y(t')$ multipliziert wird:

$$\begin{aligned} P_{k_B}^B(t - t') &= \sum_{i=1}^{k_B+1} g_{i,B}(t - t')^{i-1} \\ &= \sum_{i=1}^{k_B+1} g_{i,B} \sum_{j=0}^{i-1} \binom{i-1}{j} t^{i-1-j} (-1)^j (t')^j \\ &= \sum_{i=1}^{k_B+1} \underbrace{\left(\sum_{j=i}^{k_B+1} g_{j,B} \binom{j-1}{i-1} t^{j-i} (-1)^{i-1} \right)}_{=: \tilde{g}_{i,B}(t)} (t')^{i-1}. \end{aligned} \quad (6.13)$$

Die Multiplikation der beiden Polynome kann nun folgendermaßen ausgeführt werden:

$$P_{k_B}^B(t - t') \cdot P_{k_y}^y(t') = \sum_{i=1}^{k_B+k_y+1} \underbrace{\sum_{l=1}^i \tilde{g}_{l,B}(t) g_{i+1-l,y}(t')}_{=: \hat{g}_i} (t')^{i-1}. \quad (6.14)$$

Es gibt verschiedene Möglichkeiten, das Produkt zweier Polynome zu beschleunigen, allerdings sind diese Verfahren für Polynome mit niedrigen Ordnungen (es werden keine Polynome mit Ordnung größer als zwölf aufgestellt) nicht effizienter als das naive Ausmultiplizieren [2].

Als letzten Schritt einer Funktionsauswertung benötigen wir das Integral des Produktes von $P_p^y(t')$ und $P_m^B(t - t')$ auf dem Intervall $[t_l, t_r]$, wobei die korrekte Wahl von t_l und t_r am Ende von Kapitel 6.1 näher erläutert wird. Dieses Produkt liegt in der Form (6.12) vor, weshalb sich eine Quadratur anbietet:

$$\int_{t_l}^{t_r} \sum_{i=1}^{k+1} \hat{g}_i(t')^{i-1} dx = \sum_{i=1}^{k+1} \frac{\hat{g}_i}{i} ((t_r^i - t_l^i)) \quad (6.15)$$

Auf diese Weise lässt sich also das Integral in jedem Schritt berechnen. Dieser Ansatz ist analytisch so einfach wie möglich, allerdings ergibt sich ein grundsätzliches Problem in der numerischen Stabilität dieser Variante. Dieses wird in Kapitel 7 genauer erläutert. Um die numerische Instabilität zu minimieren wird, eine weitere Variante eingeführt, die größeren Fokus auf numerische Stabilität setzt.

6.4 Multiplikation und Integration, Variante zwei

Die Idee der zweiten Variante ist es, an Stelle von Polynomen der Form (6.12), direkt Polynome der Form (6.9) zu multiplizieren und integrieren. Dadurch werden Größen verwendet, die auch im ABM113 benutzt werden und deren Multiplikation und Addition wesentlich stabiler ist, da die Werte so zusammengefasst werden, dass sie möglichst in der gleichen Größenordnung liegen. Das verwendete Verfahren zur Integration ist dem Verfahren nachempfunden, das beim ABM113 zur Interpolation von Zwischenpunkten benutzt wird, vergleiche dazu [11].

- Speicherung der Vergangenheit

Die Interpolationspolynome der Vergangenheit für $y(t)$ werden anders als vorher nicht verändert, es werden direkt die $\Psi_i(n)$ und die $\Phi_i(n)$ eines n ten Schrittes für $i = 1, \dots, k$ sowie die Ordnung k gespeichert. Bei den Interpolationspolynomen für $B(t - t', y(t - t'))$ werden die Ψ noch ein wenig modifiziert:

$$\begin{aligned} P_{k_B, n}^B(t - t') &= \sum_{i=1}^{k_B+1} \Phi_i^B \prod_{m=1}^{i-1} \frac{t - t' - x_n^B + \Psi_{m-1}^B}{\Psi_m^B} \\ &= \sum_{i=1}^{k_B+1} \Phi_i^B \prod_{m=1}^{i-1} \frac{t' + x_n^B - t + \tilde{\Psi}_{m-1}^B}{-\tilde{\Psi}_m^B}. \end{aligned} \quad (6.16)$$

Es bietet sich also an, bei der Speicherung direkt $\tilde{\Psi}^B = (-1) \cdot \Psi^B$ an Stelle von Ψ^B zu speichern, womit sich (6.16) zu

$$P_{k_B, n}^B(t - t') = \sum_{i=1}^{k_B+1} \Phi_i^B \prod_{m=1}^{i-1} \frac{t' + x_n^B - t + \tilde{\Psi}_{m-1}^B}{\tilde{\Psi}_m^B} \quad (6.17)$$

vereinfacht.

- Multiplikation

Das Polynom (6.17) wird mit dem Interpolationspolynom für $y(t')$

$$P_{k_y, n}^y(t') = \sum_{j=1}^{k_y+1} \Phi_j^y \prod_{p=1}^{j-1} \frac{t' - x_n^y + \Psi_{p-1}^y}{\Psi_p^y} \quad (6.18)$$

multipliziert:

$$P_{k_y, n}^y(t') \cdot P_{k_B, n}^B(t - t') = \sum_{i=1}^{k_B+1} \sum_{j=1}^{k_y+1} \Phi_i^B \Phi_j^y \prod_{m=1}^{i-1} \frac{t' + x_n^B - t + \tilde{\Psi}_{m-1}^B}{\tilde{\Psi}_m^B} \prod_{p=1}^{j-1} \frac{t' - x_n^y + \Psi_{p-1}^y}{\Psi_p^y}. \quad (6.19)$$

- Integration

Für die Integration muss zuerst noch eine Vorüberlegung angestellt werden: Ein Integral

$$\int_{t_l}^{t_r} f(x) dx \quad (6.20)$$

kann durch die Substitution $x(s) = t_l + s(t_r - t_l)$, also $\frac{d}{ds}x(s) = t_r - t_l$, $s(x) = \frac{x-t_l}{t_r-t_l}$, $s(t_l) = 0$, $s(t_r) = 1$, und die Bezeichnung $h = t_r - t_l$ zu

$$\int_0^1 f(t_l + s(t_r - t_l))(t_r - t_l)ds = h \int_0^1 f(t_l + sh)ds \quad (6.21)$$

umgeformt werden. Das Integral von t_l bis t_r des Produkts (6.19) lässt sich also als

$$h \cdot \sum_{i=1}^{k_B+1} \sum_{j=1}^{k_y+1} \Phi_i^B \Phi_j^y \int_0^1 \prod_{m=1}^{i-1} \frac{t_l + sh + x_n^B - t + \tilde{\Psi}_{m-1}^B}{\tilde{\Psi}_m^B} \prod_{p=1}^{j-1} \frac{t_l + sh - x_n^y + \Psi_{p-1}^y}{\Psi_p^y} ds \quad (6.22)$$

schreiben. Die Bestimmung von t_l und t_r wird am Ende von Kapitel 6.1 erläutert. Aus Gründen der Übersicht werden für $m = 2, \dots, k_B$ und $p = 2, \dots, k_y$ weitere Zusammenfassungen durchgeführt:

$$\begin{aligned} \Gamma_{1,y}(s) &:= \frac{sh - x_n^y + t_l}{\Psi_1^y} \\ \Gamma_{p,y}(s) &:= \frac{sh - x_n^y + t_l + \Psi_{p-1}^y}{\Psi_p^y} \\ \Gamma_{1,B}(s) &:= \frac{sh + x_n^B + t_l - t}{\tilde{\Psi}_1^B} \\ \Gamma_{m,B}(s) &:= \frac{sh + x_n^B + t_l - t + \tilde{\Psi}_{m-1}^B}{\tilde{\Psi}_m^B}. \end{aligned} \quad (6.23)$$

Die $\Gamma_{p,y}(s)$ und $\Gamma_{m,B}(s)$ haben für $m = 1, \dots, k_B$ und $p = 1, \dots, k_y$ folgende Eigenschaften:

$$\begin{aligned} \frac{d}{ds} \Gamma_{p,y}(s) &= \frac{h}{\Psi_p^y} =: \eta_{p,y}(s) = \eta_{p,y} \\ \frac{d}{ds} \Gamma_{m,B}(s) &= \frac{h}{\tilde{\Psi}_m^B} =: \eta_{m,B}(s) = \eta_{m,B}, \end{aligned} \quad (6.24)$$

die Ableitungen der $\Gamma_{p,y}(s)$ und $\Gamma_{m,B}(s)$ sind also nicht mehr von s abhängig. Mit (6.23) und der Substitution, die zu (6.22) führt, gilt dann

$$\int_{t_l}^{t_r} P_{k_y,n}^y(t') \cdot P_{k_B,n}^B(t - t') dt' = h \cdot \sum_{i=1}^{k_B+1} \sum_{j=1}^{k_y+1} \Phi_i^B \Phi_j^y \int_0^1 \prod_{m=1}^{i-1} \Gamma_{m,B}(s) \prod_{p=1}^{j-1} \Gamma_{p,y}(s) ds. \quad (6.25)$$

Die auftretenden Produkte der $\Gamma_{p,y}(s)$ und $\Gamma_{m,B}(s)$ lassen sich in einer Matrix

$$\begin{array}{ccccccc} & i=1 & i=2 & i=3 & \dots & & \\ j=1 & & 1 & \Gamma_{1,y} & \Gamma_{1,y}\Gamma_{2,y} & \dots & \\ j=2 & & 1 & \Gamma_{1,B} & \Gamma_{1,B}\Gamma_{1,y} & \Gamma_{1,B}\Gamma_{1,y}\Gamma_{2,y} & \dots \\ j=3 & 1 & \Gamma_{1,B} & \Gamma_{1,B}\Gamma_{2,B} & \Gamma_{1,B}\Gamma_{2,B}\Gamma_{1,y} & \Gamma_{1,B}\Gamma_{2,B}\Gamma_{1,y}\Gamma_{2,y} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \end{array} \quad (6.26)$$

veranschaulichen. Dabei sind die Spalten vor $i = 1$ so aufgefüllt worden, dass die Produkte immer mit eins beginnen. Diese Darstellung erleichtert es, im Folgenden die Vorgehensweise nachzuvollziehen.

Um die Lesbarkeit zu erhöhen wird nur das Integral der Terme für $j = 1$, also derjenigen, die nur $\Gamma_{p,y}(s)$ enthalten, detailliert berechnet und anschließend auf $j = 2, \dots, k_B$ verallgemeinert. Dies ist möglich, da die Struktur der $\Gamma_{p,y}(s)$ und $\Gamma_{m,B}(s)$ identisch ist und die Zeilen für $j = 2, \dots, k_B$

lediglich mehr Faktoren aufweisen. Außerdem wird der Exponent $(-q)$ als Bezeichnung für die q te Stammfunktion einer Funktion benutzt, für $q = 1$ gilt also beispielsweise

$$\int f(x)dx = f^{(-1)}(x). \quad (6.27)$$

Für ein beliebiges fixiertes j^* lässt sich dann durch partielle Integration die erste Stammfunktion von $j^* - 1$ Faktoren auf die erste und zweite Stammfunktion von $j^* - 2$ Faktoren zurückführen:

$$\begin{aligned} \left(\prod_{p=1}^{j^*-1} \Gamma_{p,y}(s) \right)^{(-1)} &= \left(\prod_{p=1}^{j^*-2} \Gamma_{p,y}(s) \Gamma_{j^*-1,y}(s) \right)^{(-1)} \\ &= \left(\prod_{p=1}^{j^*-2} \Gamma_{p,y}(s) \right)^{(-1)} \Gamma_{j^*-1,y}(s) - \left(\left(\prod_{p=1}^{j^*-2} \Gamma_{p,y}(s) \right)^{(-1)} \frac{d}{ds} \Gamma_{j^*-1,y}(s) \right)^{(-1)} \\ &= \left(\prod_{p=1}^{j^*-2} \Gamma_{p,y}(s) \right)^{(-1)} \Gamma_{j^*-1,y}(s) - \eta_{j^*-1,y} \left(\prod_{p=1}^{j^*-2} \Gamma_{p,y}(s) \right)^{(-2)}. \end{aligned} \quad (6.28)$$

Dieses Verfahren lässt sich wiederholt durchführen, bis durch die partielle Integration nur noch der Faktor für $j^* - 1 = 1$ übrig ist. Dabei müssen in jedem weiteren Schritt n_I die erste bis zur n_I ten Stammfunktion eines Produktes mit einem Faktor weniger als im Schritt zuvor berechnet werden. Im letzten Schritt werden dann die erste bis $(j^* - 1)$ te Stammfunktion des letzten übrig gebliebenen Faktors benötigt. Diese Stammfunktionen lassen sich wieder durch partielle Integration auf Stammfunktionen von s zurückführen, die sich gut berechnen lassen und die vor allem in den Grenzen von 0 bis 1 sehr einfach werden. Zur Verdeutlichung wird

$$\int \Gamma_{1,y} \Gamma_{2,y} ds \quad (6.29)$$

nach diesem Verfahren berechnet:

$$\begin{aligned} (\Gamma_{1,y} \Gamma_{2,y})^{(-1)} &= (\Gamma_{1,y})^{(-1)} \Gamma_{2,y} - \eta_{2,y} (\Gamma_{1,y})^{(-2)} \\ &= [(1)^{(-1)} \Gamma_{1,y} - \eta_{1,y} (1)^{(-2)}] \Gamma_{2,y} - \eta_{2,y} [(1)^{(-1)} \Gamma_{1,y} - \eta_{1,y} (1)^{(-2)}]^{(-1)} \\ &= [s \Gamma_{1,y} - \eta_{1,y} \frac{1}{2} s^2] \Gamma_{2,y} - \eta_{2,y} [(s \Gamma_{1,y})^{(-1)} - \eta_{1,y} (\frac{1}{2} s^2)^{(-1)}] \\ &= [s \Gamma_{1,y} - \eta_{1,y} \frac{1}{2} s^2] \Gamma_{2,y} - \eta_{2,y} [(s^{(-1)} \Gamma_{1,y} - \eta_{1,y} s^{(-2)}) - \eta_{1,y} \frac{1}{6} s^3] \\ &= [s \Gamma_{1,y} - \eta_{1,y} \frac{1}{2} s^2] \Gamma_{2,y} - \eta_{2,y} [(\frac{1}{2} s^2 \Gamma_{1,y} - \eta_{1,y} \frac{1}{6} s^3) - \eta_{1,y} \frac{1}{6} s^3] \\ &= [s \Gamma_{1,y} - \eta_{1,y} \frac{1}{2} s^2] \Gamma_{2,y} - \eta_{2,y} [(\frac{1}{2} s^2 \Gamma_{1,y} - \eta_{1,y} \frac{1}{3} s^3)] \end{aligned} \quad (6.30)$$

Wenn man das ganze Verfahren nun umdreht und mit den Stammfunktionen von s beginnt, lässt sich eine Rekursionsformel finden, mit der sich die erste Stammfunktion der Produkte (6.26) relativ einfach berechnen lässt. Die höheren Stammfunktionen werden dabei in jedem Schritt nur berechnet, um die Stammfunktionen im nächsten Schritt berechnen zu können. Bei der Rekursionsformel ist bereits berücksichtigt, dass das Einsetzen von Null in eine Stammfunktion von s beliebiger Höhe Null ergibt. Die $g_{i,q}$ der Rekursionsformel entsprechen der q ten

Stammfunktion des Produktes aus $j - 1$ Faktoren $\Gamma_{p,y}(s)$ multipliziert mit $(q - 1)!$:

$$g_{i,q} = (q - 1)! \underbrace{\int_0^1 \dots \int_0^1}_{q \text{ mal}} \prod_{p=1}^{i-1} \Gamma_{p,y}(s) d^q s = \begin{cases} \frac{1}{q} & \text{für } i = 1 \\ \Gamma_{i-1}(1)g_{i-1,q} - \eta_{i-1}g_{i-1,q+1} & \text{für } i \geq 2, \end{cases} \quad (6.31)$$

$g_{i,1}$ entspricht also der ersten Stammfunktion der entsprechenden Produkte von $\Gamma_{p,y}(s)$.

Diese Rekursionsformel wird zur Veranschaulichung für $i = 3$ ausgeführt:

$$\begin{array}{lll} & q = 1 & q = 2 \quad q = 3 \\ i = 1 & \frac{1}{1} & \frac{1}{2} \quad \frac{1}{3} \\ i = 2 & \Gamma_{1,y} - \eta_{1,y} \frac{1}{2} & \Gamma_{1,y} \frac{1}{2} - \eta_{1,y} \frac{1}{3} \\ i = 3 & \Gamma_{2,y} [\Gamma_{1,y} - \eta_{1,y} \frac{1}{2}] - \eta_{2,y} [\Gamma_{1,y} \frac{1}{2} - \eta_{1,y} \frac{1}{3}]. & \end{array} \quad (6.32)$$

Es ergibt sich also das Gleiche wie in (6.30) für $s = 1$. Der Fall $s = 0$ fällt, wie weiter oben erläutert, weg. Dieses Verfahren wird dann zeilenweise auf (6.26) angewendet, wobei das Verfahren in jeder Zeile neu gestartet wird und somit in der j ten Zeile nur die $g_{i+j,1}$ relevant sind. Es ergibt sich also für das Integral von (6.19) insgesamt:

$$\int_{t_l}^{t_r} P_{k_y,n}^y(t') \cdot P_{k_B,n}^B(t - t') dt' = h \cdot \sum_{j=1}^{k_y+1} \sum_{i=1}^{k_B+1} \Phi_i^B \Phi_j^y g_{i+j,1}. \quad (6.33)$$

Diese Doppelsumme muss dann nur noch ausgeführt werden.

7 Ergebnisse

Im Rahmen dieses Kapitels werden drei Memory-Kerne untersucht und die Ergebnisse von drei Solvern ausgeführt. Dabei werden ein Solver mit konstanter Schrittweite, der ein Prädiktor-Korrektor-Verfahren vierter Ordnung mit einer zusammengesetzten Trapezregel kombiniert (ABM4), der Solver, der am ITP-Bremen realisiert und in Kapitel 4 erläutert wurde (ABMitp) und der in Kapitel 5 und 6 entwickelte ABM113 in Variante zwei, ABM113v2, verwendet.

7.1 Konstanter Memory-Kern

- Analytische Betrachtungen

Als erstes wird der Kern $B(t-t', y(t-t')) = -1$ betrachtet, das zugehörige Anfangswertproblem lautet

$$\begin{aligned}\frac{d}{dt}y(t) &= \int_0^t -1 \cdot y(t') dt' \\ y(0) &= 1.\end{aligned}\tag{7.1}$$

Streng genommen ist dieser Kern kein Memory-Kern, da er überhaupt nicht zeitabhängig ist, aber für Testzwecke ist er sehr gut geeignet, da sich das Anfangswertproblem analytisch lösen lässt und die Lösungsfunktion sehr einfach ist. Wenn man beide Seiten der Gleichung ableitet,

$$\frac{d^2}{dt^2}y(t) = -\frac{d}{dt} \int_0^t y(t') dt',\tag{7.2}$$

lässt sich (7.1) mit dem Hauptsatz der Differential- und Integralrechnung (HSDIR) zu

$$\begin{aligned}\frac{d^2}{dt^2}y(t) &= -y(t) \\ y(0) &= 1\end{aligned}\tag{7.3}$$

umformen. Wenn man zusätzlich eine Anfangsbedingung $y'(0) = 0$ annimmt, dann ergibt sich:

$$y(t) = \cos(t).\tag{7.4}$$

- Numerische Betrachtungen

Dieses Anfangswertproblem lässt sich mit allen Solvern sinnvoll lösen, natürlicherweise wird die Rechnung auf einem konstanten Gitter für größere Endzeitpunkte länger dauern, dieses Verhalten wurde in Kapitel 1.2 erläutert und wird in Kapitel 7.3 erneut aufgegriffen. In Abbildung 5 ist zu sehen, dass der ABM113v2 für die Lösung dieses Problems sehr wenige Schritte benötigt und die Funktion $y(t) = \cos(t)$ gut interpoliert.

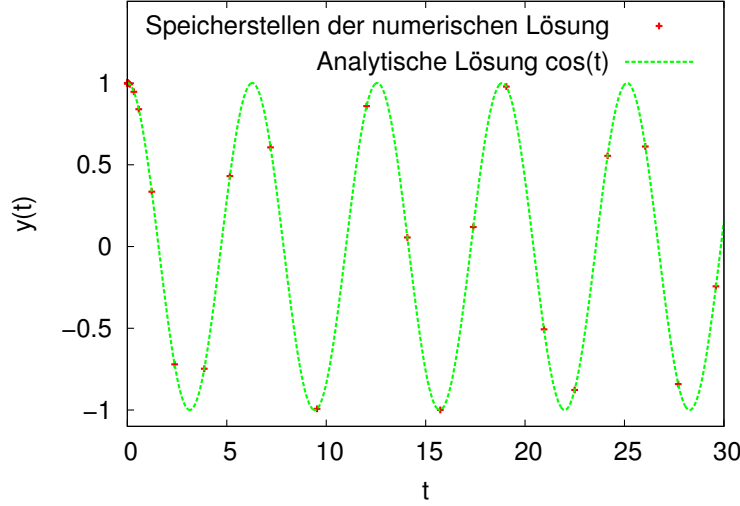


Abbildung 5: Hier ist die Lösung der Integro-Differentialgleichung mit dem Memory-Kern $B(t - t', y(t - t')) = -1$ abgebildet. Dargestellt ist die analytische Lösung und die numerisch berechneten Funktionswerte von $y(t)$ an den Speicherpunkten der Polynome des ABM113v2, die Rechnung des ABM113v2 erfolgte mit einer geforderten Genauigkeit von $\eta = 10^{-7}$.

7.2 Linearer Memory-Kern

Der lineare Memory-Kern $B(t - t', y(t - t')) = -(t - t')$, der auf das Anfangswertproblem

$$\begin{aligned} \frac{d}{dt}y(t) &= \int_0^t -(t - t')y(t')dt' \\ y(0) &= 1 \end{aligned} \quad (7.5)$$

führt, lässt sich ebenfalls noch analytisch lösen und ist das einfachste Beispiel für einen zeitabhängigen Memory-Kern.

- Analytische Betrachtungen

Die Integro-Differentialgleichung (7.5) lässt sich zu

$$\frac{d}{dt}y(t) = -t \int_0^t y(t')dt' + \int_0^t t'y(t')dt' \quad (7.6)$$

umformen. Wie beim konstanten Memory-Kern lässt sich diese Gleichung durch wiederholtes Ableiten beider Seiten und dem Einsatz des HSDIR lösen:

$$\begin{aligned} \frac{d^2}{dt^2}y(t) &= -\int_0^t y(t')dt' - t \frac{d}{dt} \int_0^t y(t')dt' + \frac{d}{dt} \int_0^t t'y(t')dt' \\ &= -\int_0^t y(t')dt' - ty(t) + ty(t) \\ &= -\int_0^t y(t')dt' \\ \frac{d^3}{dt^3}y(t) &= -y(t). \end{aligned} \quad (7.7)$$

Das resultierende Anfangswertproblem

$$\begin{aligned}\frac{d^3}{dt^3}y(t) &= -y(t) \\ y(0) &= 1\end{aligned}\tag{7.8}$$

hat mit den zusätzlichen Annahmen

$$\begin{aligned}y'(0) &= 0 \\ y''(0) &= 0\end{aligned}\tag{7.9}$$

die Lösung

$$y(t) = \frac{1}{3}e^{-t} + \frac{2}{3}e^{\frac{t}{2}}\cos\left(\frac{\sqrt{3}}{2}t\right).\tag{7.10}$$

- Numerische Betrachtungen

Auch dieses Testproblem lässt sich für die verwendeten Solver ABM4, ABMitp und ABM113v2 sinnvoll lösen. Als einfachstes Beispiel für einen zeitabhängigen Memory-Kern eignet sich dieser Memory-Kern besonders zum Testen der in Kapitel 6.2 beschriebenen Auswahl der richtigen Interpolationspolynome für $B(t - t', y(t - t'))$ im Verlaufe der Multiplikation. Der konstante Memory-Kern ist hierfür nicht geeignet, da er sich beim Spiegeln und verschieben nicht verändert. Außerdem ist der lineare Memory-Kern ausreichend komplex, um die numerische Instabilität der ersten Variante der Multiplikation und Integration im ABM113, ABM113v1, die in Kapitel 6.3 beschrieben wird, zu erläutern. Die Polynome, die im ABM113v1 zur Interpolation von $\frac{d}{dt}y(t)$ aufgebaut werden, sind von der Form (6.9):

$$P_k(x) = \sum_{i=1}^k \Phi_i \prod_{m=1}^{i-1} \frac{x - x_n + \Psi_{m-1}}{\Psi_m}.\tag{7.11}$$

Diese Φ_i und Ψ_m werden, wie in (6.10) beschrieben, zusammengefasst:

$$d_i = \frac{\Phi(i)}{\prod_{l=1}^{i-1} \Psi_l} \text{ und } c_m = \begin{cases} -x_n & \text{für } m = 1 \\ -x_n + \Psi_{m-1} & \text{für } m > 1 \end{cases}\tag{7.12}$$

Schon dabei ist zu sehen, dass das Produkt der Ψ_m , welches ein Produkt von möglicherweise kleinen Schrittweiten ist, sehr klein und die d_i , die den Kehrwert dieses Produkte enthalten, somit sehr groß wird. Aber selbst wenn man versucht, die wesentlich stabileren Größen

$$d_i = \Phi(i) \text{ und } c_m = \begin{cases} \frac{-x_n}{\Psi_m} & \text{für } m = 1 \\ \frac{-x_n + \Psi_{m-1}}{\Psi_m} & \text{für } m > 1 \end{cases}\tag{7.13}$$

zu bilden, entstehen durch die Anwendung des inversen Hornerchemas Koeffizienten, die sehr groß werden. Beim unvermeidlichen Ausmultiplizieren von

$$\prod_{m=1}^{i-1} \frac{x - x_n + \Psi_{m-1}}{\Psi_m}\tag{7.14}$$

entstehen Summen von Produkten von c_m und d_i , die für den einfachen Fall des linearen Memory-Kerns und für ein Polynom von zwölfter Ordnung schon Koeffizienten g_i , die wie in (6.12) die Koeffizienten eines Polynoms der Form

$$P_k(x) = \sum_{i=1}^{k+1} g_i x^{i-1} \quad (7.15)$$

bezeichnen, in der Größenordnung 10^{16} erzeugen. Wenn diese anschließend nach dem Schema (6.14) mit einem anderen Polynom der selben Form multipliziert werden, entstehen Polynome der Form (6.12) mit maximaler Ordnung 24. Diese Polynome mit einer Quadratur (6.15) zu behandeln, führt im schlechtesten Fall dazu, dass ein Term der Form $\frac{\tilde{g}_{24}}{25} x^{25}$ in den Grenzen eines kleinen Intervalls ausgewertet wird. Dabei werden dann zwei sehr große Zahlen voneinander abgezogen was Auslöschungen unumgänglich macht. Dieses Verfahren führt für den linearen Memory-Kern etwa bei $t = 8$ zu so großen Auslöschungen, dass der Solver divergiert. Verfahren die Polynome erst in die Form (6.12) umbauen führen immer auf Probleme dieser Art, wenn höhere Polynomordnungen erreicht werden.

Eine Möglichkeit, dieses Verfahren trotzdem noch anwenden zu können besteht darin, die maximale Ordnung auf etwa vier zu begrenzen. Dann sind die Koeffizienten g_i und Exponenten in der Integration noch so klein, dass ausreichend wenig Auslöschung auftritt. Allerdings werden die pro Polynom abgedeckten Intervalle auch um $\frac{2}{3}$ kleiner, was den Gesamtaufwand, der bei jeder Integration anfällt, ungefähr verdreifacht.

7.3 Physikalischer Memory-Kern: Polaron-Selbstenergie

Wie in Kapitel 3 beschrieben entspricht der Memory-Kern der Dyson-Gleichung

$$\begin{aligned} B(t-t', y(t-t')) &= \hat{S}_\alpha^r(\tau-\tau') \cdot e^{-\frac{i}{\hbar}\varepsilon_\alpha(\tau-\tau')} \\ &= \sum_{q,\beta} |M_q^{\alpha\beta}|^2 \left((1 + N_{LO}) e^{-i\omega_{LO}(\tau-\tau')} + N_{LO} e^{i\omega_{LO}(\tau-\tau')} \right) \\ &\cdot e^{-\frac{\Gamma_{LO}}{\hbar}(\tau-\tau')} \tilde{G}_\beta^r(\tau-\tau') e^{-\frac{i}{\hbar}\varepsilon_\beta(\tau-\tau')} \cdot e^{-\frac{i}{\hbar}\varepsilon_\alpha(\tau-\tau')}. \end{aligned} \quad (7.16)$$

Dabei ist zu beachten, dass der Wetting-Layer bei der Berechnung der Matrixelemente $M_q^{\alpha\beta}$ so genähert wurde, dass er nur noch von $|k|$ abhängt. Dieses $|k|$ wird mit einem äquidistanten Gitter aus "k-Punkten" diskretisiert. Außerdem werden Genauigkeiten η von Solvern verglichen, wobei nicht zwischen relativen und absoluten Fehlern unterschieden wird, auch wenn die adaptiven Solver dies ermöglichen. Es wird also immer $\eta_{abs} = \eta_{rel} =: \eta$ angenommen.

Die Zeitentwicklung des Memory-Kerns der Dyson-Gleichung ist in Abbildung 6 für die s-Schale und den ersten Wetting-Layer Zustand abgebildet, in Abbildung 7 ist die zugehörige Zeitentwicklung der Lösung der Dyson-Gleichung zu sehen.

Zu beachten ist, dass die Multiplikation der Graphen der beiden Polynome von $B(t-t', y(t-t'))$ und $y(t')$ nicht wie in Abbildung 6 und 7 dargestellt erfolgt, sondern der Graph von $B(t-t')y(t-t')$ erst noch, wie in Kapitel 6.2 beschrieben, auf dem Intervall $[0, t]$ gedreht werden muss. Vor diesem Hintergrund ist das in Kapitel 6.1 beschriebene Verhalten zu erkennen, dass die größte Änderung von $y(t')$ zu Beginn und die von $B(t-t', y(t-t'))$ am Ende der Integration über t' erfolgt.

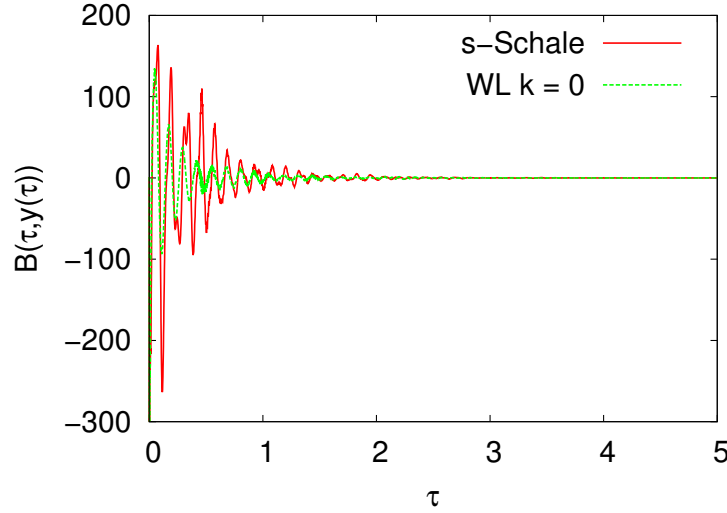


Abbildung 6: Zeitentwicklung des Memory-Kerns der Dyson-Gleichung, es ist die s-Schale und der erste Wetting-Layer Zustand zu sehen. Die Rechnung erfolgte mit 120 k-Punkten bei einer geforderten Genauigkeit von $\eta = 10^{-6}$ und einer Temperatur von 300K bis zu einem Endzeitpunkt von 30 ps.

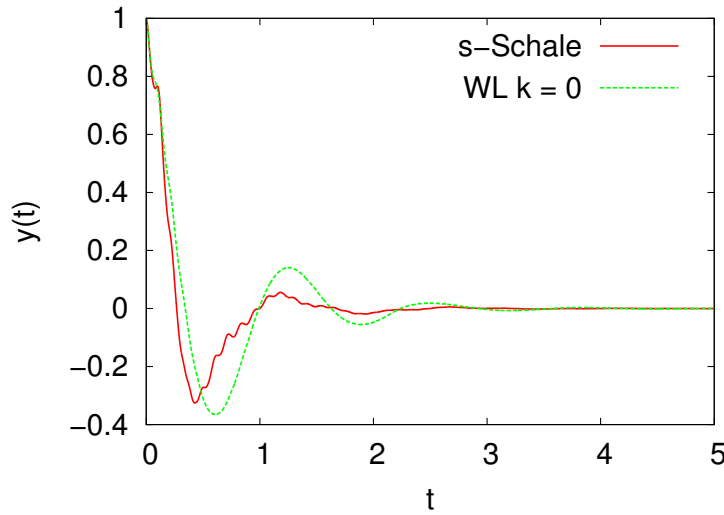


Abbildung 7: Zeitentwicklung der numerischen Lösung der Dyson-Gleichung, es ist die s-Schale und der erste Wetting-Layer Zustand abgebildet. Die Daten der Abbildung entstammen der gleichen Rechnung wie Abbildung 6.

Physikalisch interessant sind allerdings nicht die Zeitentwicklungen direkt, sondern die Fouriertransformation (3.12), $\hat{G}_k(\omega)$, von der Zeitentwicklung der Lösung $y(t)$. Im Folgenden wird diese fouriertransformierte Zeitentwicklung genauer untersucht und diskutiert. Dafür werden zunächst für eine feste Anzahl von k-Punkten und fester Temperatur verschiedene geforderte Genauigkeiten betrachtet, anschließend bei fester Genauigkeit verschiedene k-Gitter und Temperaturen betrachtet und schließlich für ein festes Gitter, eine Temperatur und eine Genauigkeit die Leistung der drei Solver (ABM4, ABMitp und ABM113v2) verglichen.

7.3.1 120 k-Punkte, 300 Kelvin, variierte Genauigkeit

Die Entscheidung, die Genauigkeit bei 120 k-Punkten zu variieren, ist deshalb gefallen, weil für 120 k-Punkte auch eine Genauigkeit von $\eta = 10^{-7}$ betrachtet werden kann und die Spektralfunktionen für alle betrachteten Genauigkeiten ausreichend gut aussehen. 300 Kelvin ist eine übliche Annahme für die Dyson-Gleichung, weil dabei numerisch noch relativ einfach zu lösende Gleichungen entstehen. In Abbildung 8 lässt sich sehr gut erkennen, dass die geforderte Genauigkeit im Spektrum in etwa eingehalten wird. Die geforderte Genauigkeit ist als horizontale Linie in den Abbildungen dargestellt. Die nicht physikalischen zusätzlichen schnellen Oszillationen befinden sich in allen Abbildungen unterhalb dieser Linie oder - im Fall $\eta = 10^{-4}$ - nur sehr leicht darüber.

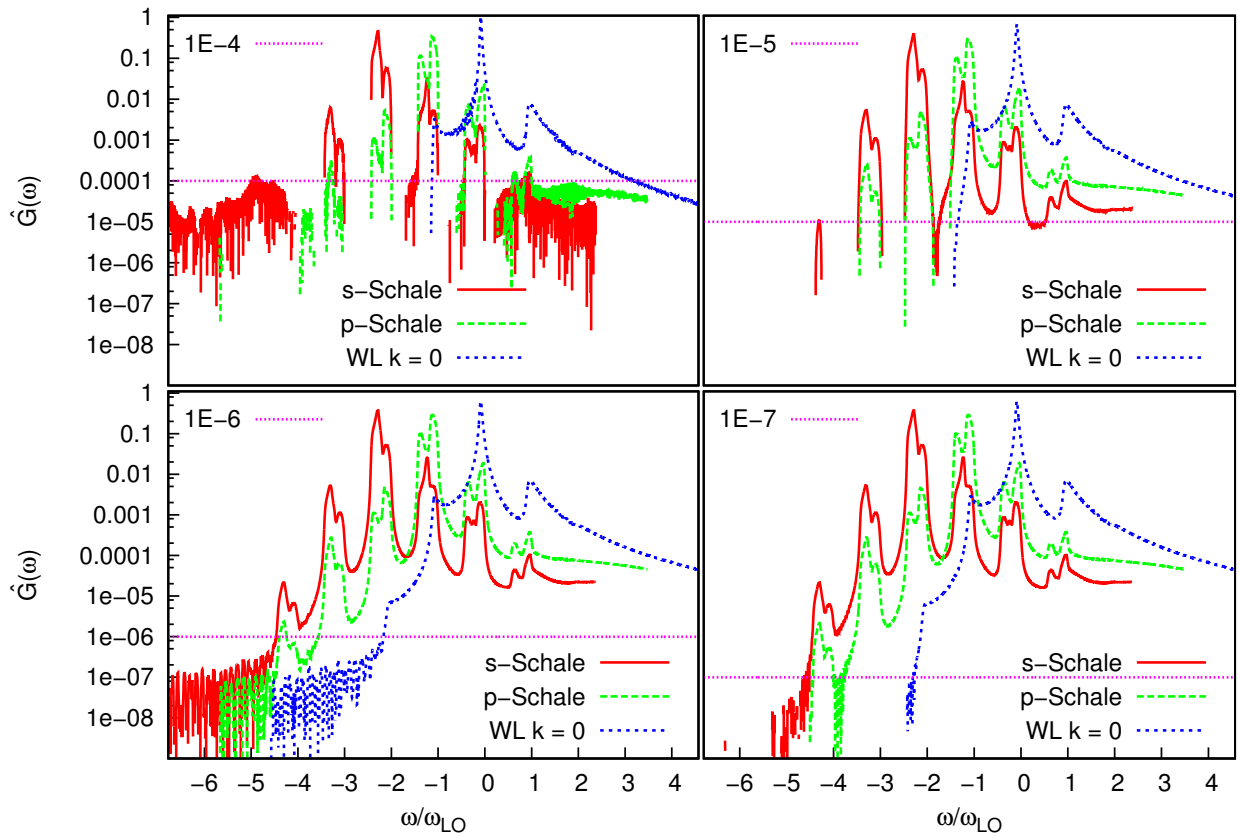


Abbildung 8: Darstellung der Abhängigkeit der Spektralfunktionen von der geforderten Genauigkeit. Die Rechnungen erfolgten bei einer Temperatur von 300K und einem k-Gitter mit 120 k-Punkten bis zu einem Endzeitpunkt von 30 ps.

Eine Genauigkeit von $\eta = 10^{-8}$ ist nicht mehr sinnvoll zu berechnen, da durch das gewählte k-Gitter (und die Genauigkeit der physikalischen Größen) eine maximal mögliche Genauigkeit entsteht, die mit $\eta = 10^{-8}$ unterschritten wird. Unterhalb von $\eta = 10^{-7}$ versucht der ABM113v2 die numerischen Fehler, die bei der Diskretisierung des Wetting-Layer entstehen, aufzulösen und wählt deshalb viel zu feine Schrittweiten. Die Wahl von wesentlich mehr k-Punkten ist nicht ohne weiteres möglich, da bei der Berechnung der Matrixelementen $M_q^{\alpha\beta}$ - zumindest auf einem gewöhnlichen Arbeitsplatzrechner - Speicherprobleme entstehen. Bei einem vorhandenen

Arbeitsspeicher von vier GB lassen sich etwa bis zu 150 k-Punkte realisieren, für 500 k-Punkte werden hingegen schon etwa 24 GB benötigt. Allerdings sind auch 500 k-Punkte noch nicht geeignet, um eine Rechnung mit geforderter Genauigkeit von $\eta = 10^{-8}$ sinnvoll durchzuführen, für 0.15% einer Zeitrechnung von 30ps werden bereits 5500 Interpolationspolynome für $y(t)$ gespeichert, was vermuten lässt, dass Oszillationen aufgelöst werden, die nicht vorhanden sein sollten.

7.3.2 $\eta = 10^{-6}$, 300 Kelvin, variierte Anzahl von k-Punkten

Als nächstes soll die Abhängigkeit der Lösung von der Anzahl der k-Punkte näher untersucht werden. Hierbei wurde eine geforderte Genauigkeit von $\eta = 10^{-6}$ gewählt, da bei dieser Genauigkeit auch ein Gitter von 60 k-Punkten noch zu einem Ergebnis führt. Außerdem wird aus den gleichen Gründen wie oben eine Temperatur von 300 Kelvin gesetzt.

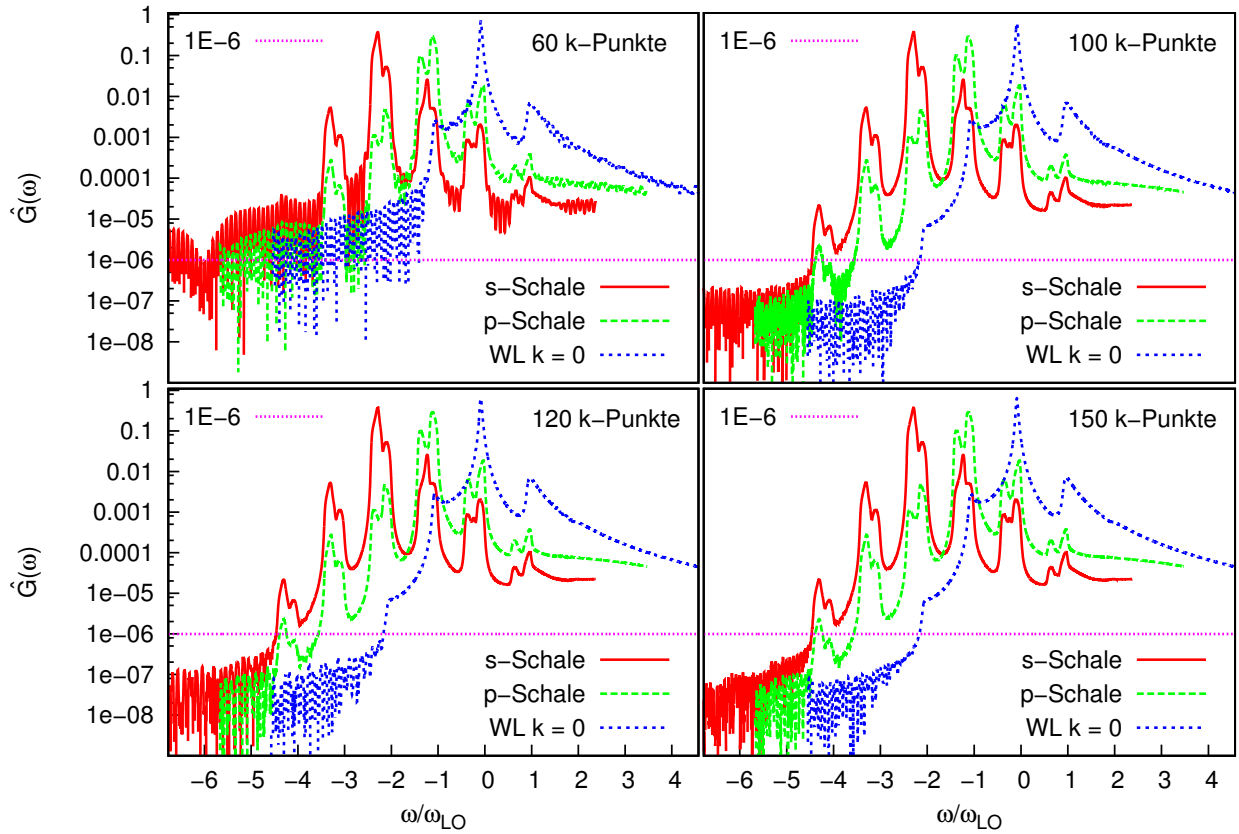


Abbildung 9: Darstellung der Abhängigkeit der Spektralfunktionen von der Anzahl der gewählten k-Punkte. Die Rechnungen erfolgten bei einer Temperatur von 300k und einer geforderten Genauigkeit von $\eta = 10^{-6}$ bis zu einem Endzeitpunkt von 30 ps.

In Abbildung 9 ist oben links zu erkennen, dass 60 k-Punkte eine zu geringe Auflösung des Wetting-Layer bieten um eine Genauigkeit von $\eta = 10^{-6}$ zu fordern, da die schnellen nicht physikalischen Oszillationen überhalb der horizontalen Linie liegen. Die Plots oben rechts und unten links zeigen, dass 100 k-Punkte ausreichen, um die Spitzen aufzulösen, beim Übergang zu 120 k-Punkten verschwinden oberhalb von $\eta = 10^{-6}$ lediglich Oszillationen in den Tälern

zwischen den Spitzen. Außerdem ist beim Vergleich der Plots unten links und unten rechts zu erkennen, dass 120 k-Punkte für eine geforderte Genauigkeit von $\eta = 10^{-6}$ ausreichen, da sich die Fälle für 120 und 150 k-Punkte mit bloßem Auge kaum noch unterscheiden.

Die Anzahl der benötigten gespeicherten Polynome für $y(t)$ nimmt für zunehmende Anzahl an k-Punkten stetig ab, bei 60 k-Punkten benötigt der ABM113v2 700, für 100 k-Punkte 680, für 120 k-Punkte 615 und schließlich für 150 k-Punkte 609 Interpolationspolynome. Das zeigt, dass die Lösungen für eine höhere Anzahl an k-Punkten immer glatter werden.

Allgemein lässt sich aus der Variation der Genauigkeit und der Variation der k-Punkte im Gitter des Wetting-Layer schließen, dass für eine höhere geforderte Genauigkeit mehr k-Punkte benötigt werden. Für eine Genauigkeit von $\eta = 10^{-6}$ bieten 120 k-Punkte ein gutes Ergebnis, für eine Genauigkeit von $\eta = 10^{-7}$ ist 150 k-Punkte eine annehmbare Wahl um die nicht physikalischen Oszillationen unter die geforderte Genauigkeit zu drücken. Es ist anzunehmen, dass der Wetting-Layer durch die Diskretisierung mit den k-Punkten mit einer entsprechenden Genauigkeit aufgelöst wird.

7.3.3 Variation der Temperatur

Die Wahl von Temperaturen unterhalb von 300K führt vor allem dazu, dass die Oszillationen in der Zeitentwicklung der Dyson-Gleichung wesentlich langsamer abfallen, wodurch der numerische Aufwand höher wird.

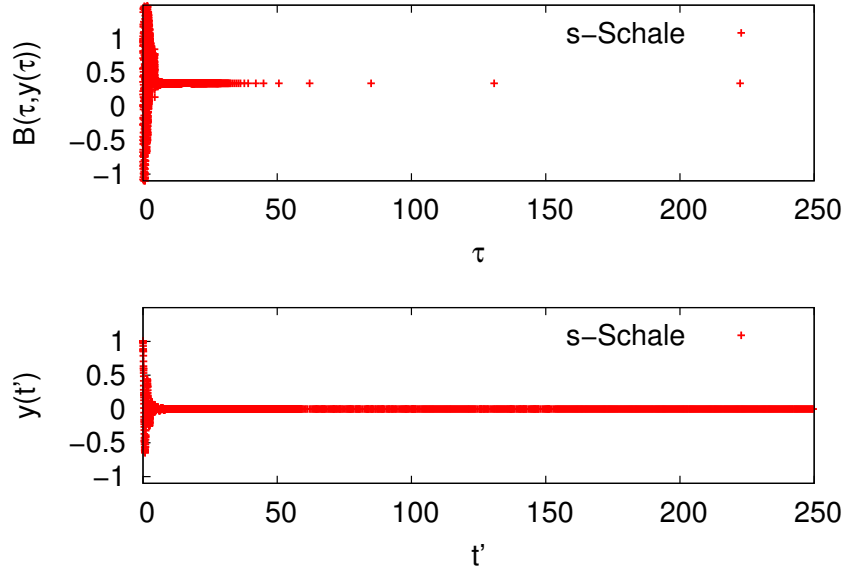


Abbildung 10: Darstellung des Speicherstellen der Interpolationspolynome von $y(t)$ im Verlauf der Propagation des ABM113v2. Die Rechnung erfolgte bei 150K, einem Gitter von 300 k-Punkten und einer geforderten Genauigkeit von $\eta = 10^{-6}$ bis zu einem Endzeitpunkt von 250 ps.

In Abbildung 10 ist zu erkennen, dass der Memory-Kern bereits bei etwa 50 ps vollständig abgefallen ist und die Oszillationen unter die geforderten Genauigkeit fallen. Das ist daran zu

erkennen, dass der ABM113v2 die Schrittweite ab diesem Zeitpunkt enorm erhöht. Die Zeitentwicklung von $y(t)$ fällt allerdings wesentlich langsamer ab und ist selbst nach 250 ps noch so stark am oszillieren, dass der ABM113v2 die Schrittweite relativ niedrig hält. Das Abfallen der Funktionswerte von $y(t)$ ist notwendig, um in den Spektralfunktionen ungewünschte Oszillationen überhalb der geforderten Genauigkeit zu unterbinden. Abbildung 11 zeigt die Spektralfunktionen dieser Rechnung. Es ist zu erkennen, dass es unsaubere Stellen oberhalb der geforderten Genauigkeit von $\eta = 10^{-6}$ gibt, die denen aus Abbildung 9 oben rechts ähneln. Das lässt vermuten, dass das gewählte k-Gitter den Wetting-Layer nicht ausreichend gut auflöst.

Eine Rechnung bei 77K ist mit 300 k-Punkten bis 250 ps nicht ausreichend abgefallen, vor allem die Zustände des Wetting-Layers sind bei diesen Temperaturen nach 250 ps nicht einmal um die Hälfte abgefallen. Es ist zu erwarten, dass dies erst nach etwa 1500-2000 ps der Fall ist.

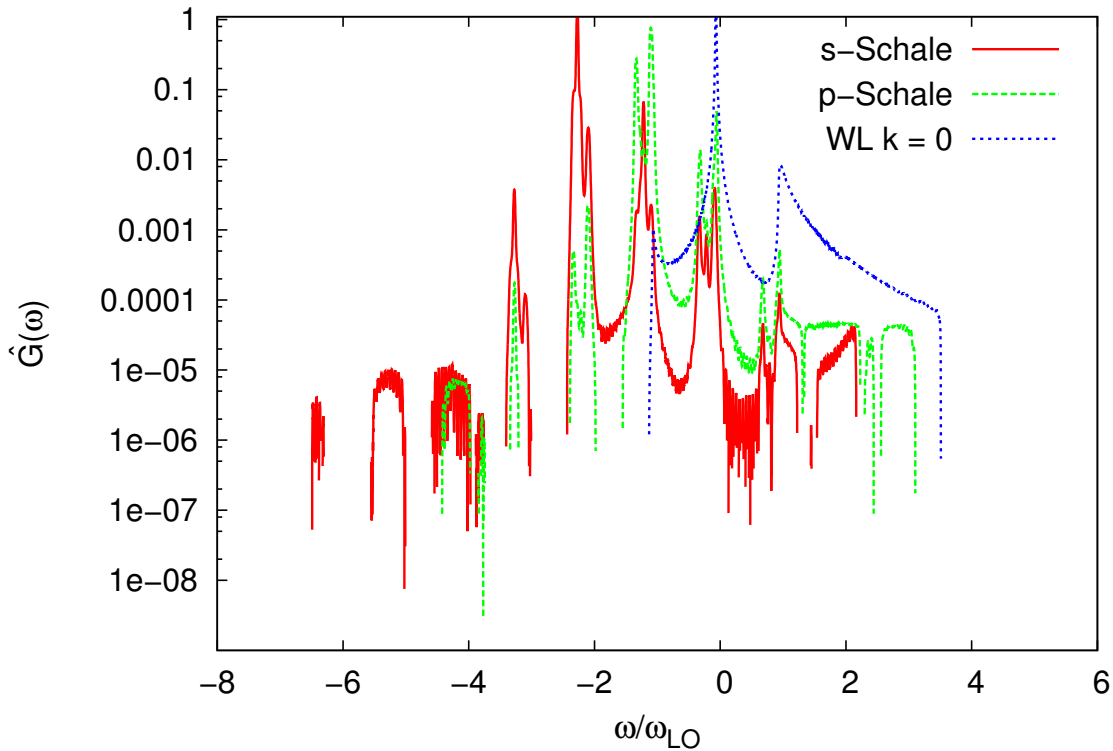


Abbildung 11: Spektralfunktionen einer Rechnung mit 150K, 300 k-Punkten und einer Genauigkeit von $\eta = 10^{-6}$ bis zu einem Endzeitpunkt von 250 ps.

7.3.4 Vergleiche der verschiedenen Solver

Als letztes sollen die Ergebnisse und die Leistung des in dieser Arbeit konstruierten Solvers ABM113v2 mit denen des ABM4 und des ABMitp verglichen werden. Die Gegenüberstellung erfolgt für 300 Kelvin, 120 k-Punkte und für die adaptiven Solver ABM113v2 und ABMitp wird eine Genauigkeit von 10^{-6} gefordert. Der ABM4 läuft auf einem Gitter von konstanter Schrittweite von einer Femtosekunde, dabei wird laufend die Genauigkeit der Lösung überprüft, diese Schrittweite erfüllt zu Beginn der Propagation, wo die größten Änderungen aufgelöst werden

müssen, etwa eine Genauigkeit von $\eta = 10^{-5}$. Gerechnet wird jeweils bis zu einem Endzeitpunkt von 30 ps. In Abbildung 12 ist zu sehen, dass die beiden adaptiven Solver gegenüber dem ABM4

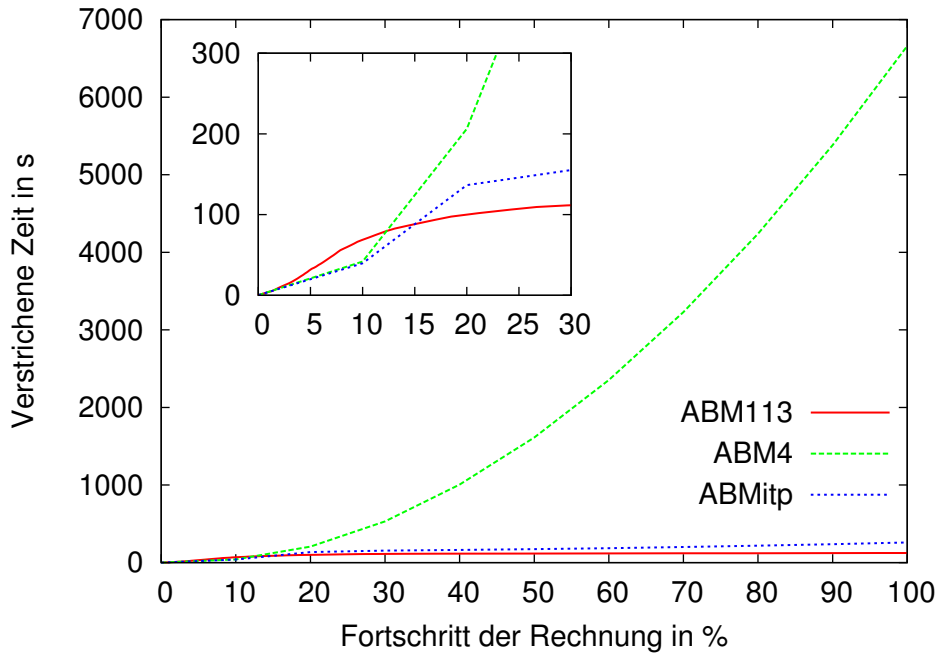


Abbildung 12: Vergleich der benötigten Zeit des ABM113v2, des ABMitp und des ABM4 im Verlaufe der Rechnung. Die Rechnung erfolgte mit 120 k-Punkten bei einer Temperatur von 300K und einer geforderten Genauigkeit von $\eta = 10^{-6}$ bis zu einem Endzeitpunkt von 30 ps. Für den ABM4 ist eine Schrittweite von einer Femtosekunde gewählt worden.

wesentlich schneller sind. Es ist für den ABM4 außerdem auch die in Kapitel 1.2 erwartete quadratische Abhängigkeit der Endzeit zur Anzahl der Punkte des Gitters zu erkennen. Im Inset der Abbildung ist der Beginn der Rechnungen vergrößert dargestellt. Dort ist zu sehen, dass die Solver ABMitp und ABM4 zu Beginn der Rechnung gegenüber dem ABM113v2 im Vorteil sind, aber nach 10 bzw. 15 % hinter der Geschwindigkeit des ABM113v2 zurückbleiben. Das liegt daran, dass zu Beginn der Rechnungen der Overhead, den der ABM113v2 erzeugt, im Verhältnis zur Berechnung des Differentials noch wesentlich mehr ins Gewicht fällt.

Es ist auch zu erkennen, dass der ABM113v2 gegenüber dem ABMitp immer besser wird, je größer der Endzeitpunkt gewählt wird. Bei der vorliegenden Rechnung benötigt der ABM113v2 bis 15 ps etwa 70% und bei 30 ps nur noch ungefähr 50% der Zeit, die der ABMitp benötigt. Dies ist damit zu erklären, dass der ABM113v2 für spätere Zeitpunkte, in denen weniger Änderungen in der gesuchten Funktion stattfinden, durch die Wahl größerer Schrittweiten nicht nur die Anzahl der Differentialauswertungen reduziert, sondern sich auch die Berechnung des Differentials beschleunigt. Diesen Vorteil hat der ABMitp nicht, da er das Differential immer auf einem festen Gitter durch eine zusammengesetzte Trapezregel bildet.

Der ABM113v2 ist aus diesem Grund besonders gut für die Aufgabe der Lösung der Dyson-Gleichung geeignet, da der gewählte Endzeitpunkt, sobald die Funktion ausreichend abgefallen ist, für die Dauer der Rechnung kaum eine Rolle spielt. Dadurch können ohne zusätzlichen Rechenaufwand sehr große Endzeitpunkte gewählt werden, um sicherzustellen, dass die Funk-

tionswerte von $y(t)$ wirklich unter die geforderte Genauigkeit abgefallen sind und dadurch in den Spektralfunktionen keine Störungen entstehen. Außerdem hat der ABM113v2 gegenüber den beiden anderen Solvern den Vorteil, dass eine geforderte Genauigkeit vorgegeben werden kann, die durch die gesamte Propagation des Solvers (natürlich nur lokal) eingehalten wird. Die Genauigkeit der zusammengesetzten Trapezregel, die der ABMitp und ABM4 verwenden, ist nur mit einigem analytischen Aufwand vorher bestimmbar und hängt dann von der Schrittweite des gewählten Gitters ab. Dadurch sind die Ergebnisse, die diese Solver liefern nicht sehr zuverlässig, was die (lokale) Genauigkeit angeht.

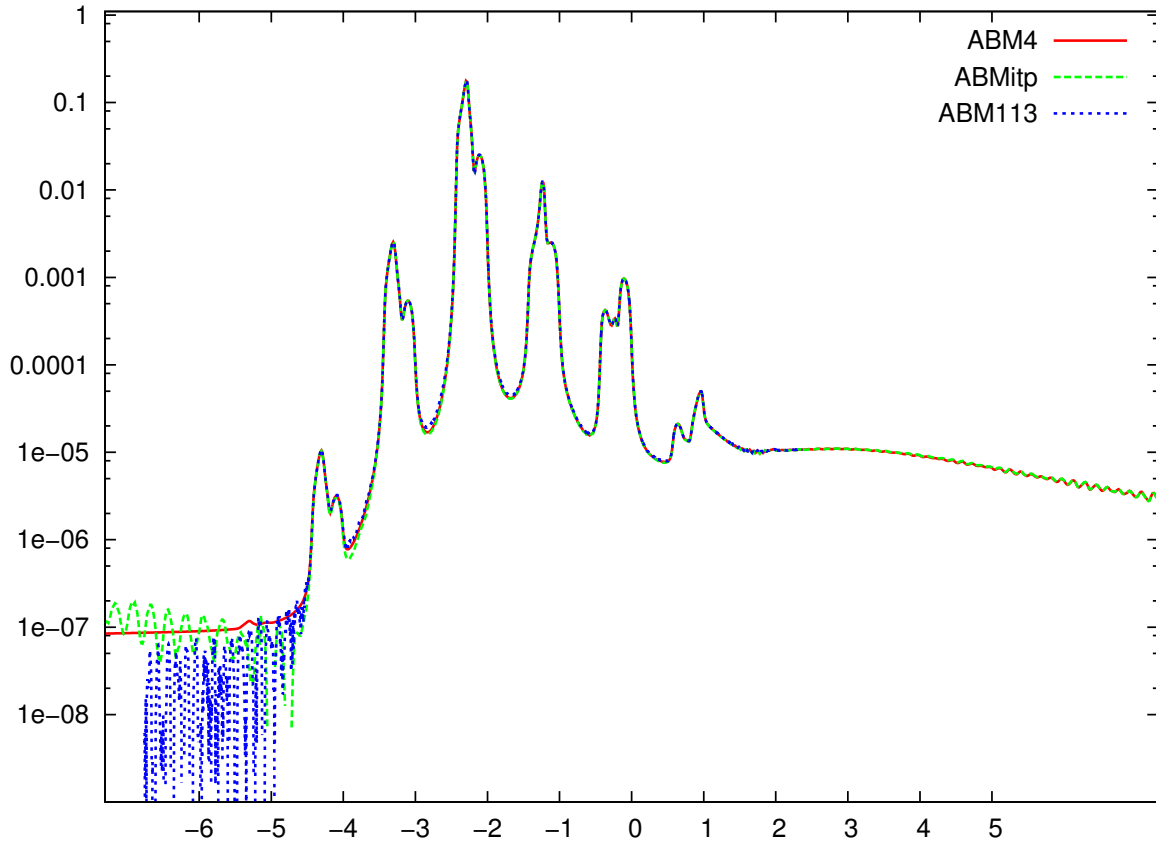


Abbildung 13: Vergleich der numerischen berechneten Spektralfunktion des ABM113v2, ABMitp und ABM4. Die Rechnung erfolgte mit 120 k-Punkten bei einer Temperatur von 300K und einer geforderten Genauigkeit von $\eta = 10^{-6}$ bis zu einem Endzeitpunkt von 30 ps. Für den ABM4 ist eine Schrittweite von einer Femtosekunde gewählt worden.

Beim Vergleich der Spektralfunktionen in Abbildung 13 zeigt sich, dass die relevanten Bereiche in der Mitte des Graphen von allen drei Solven gleich gut approximiert werden. Auffällig ist, dass an den Rändern der Spektralfunktionen, die unter der geforderten Genauigkeit von $\eta = 10^{-6}$ liegen, der Solver ABM4 augenscheinlich wesentlich bessere Ergebnisse liefert als der ABM113v2 und der ABMitp. Das liegt daran, dass der ABM4 durch das konstante Gitter, dessen Schrittweiten durch die starken Änderungen zu Beginn der Propagation bestimmt werden, das Differential zu späteren Zeitpunkten, wenn die Funktion abgefallen ist, viel genauer berechnet. Der ABM113v2 und der ABMitp lösen diese Oszillationen nur mit der geforderten Genauigkeit von $\eta = 10^{-6}$ auf und erhöhen deshalb die Schrittweite.

8 Zusammenfassung und Ausblick

Im Verlaufe dieser Arbeit wurde ein numerisches Lösungsverfahren für Integro-Differentialgleichungen entwickelt, dass eine adaptive Schrittweitensteuerung unterstützt und eine geforderte (lokale) Genauigkeit einhält. Dafür wurde das von Shampine und Gordon entwickelte Verfahren ABM113 genutzt und die innerhalb des Verfahrens aufgebauten Interpolationspolynome verwendet, um mit diesen die Lösung des Integrals zu beschleunigen. Die Leistung des Verfahrens ist durch Vergleichsrechnungen mit verschiedenen geforderten Genauigkeiten und von identischen Aufgaben mit anderen Verfahren bestätigt worden. Durch die Schrittweitensteuerung ist das Verfahren vor allem gegenüber Verfahren mit konstanten Schrittweiten ein enormer Geschwindigkeitsgewinn beim Lösen der Dyson-Gleichung für Polaronen.

Die Möglichkeit, eine (lokale) Genauigkeit fordern zu können, die während der gesamten Propagation eingehalten wird, lässt Aussagen über die (lokale) Genauigkeit der Lösung zu. Dadurch ist es möglich, optimale k-Punkt Gitter für eine gewünschte Genauigkeit zu bestimmen und einzuschätzen, wie groß der Fehler einer Lösung bei der Weiterverwendung, zum Beispiel bei der Berechnung von Streuraten, ist.

Eine interessante weiterführende Untersuchung wäre es, diese Abhängigkeit der Genauigkeit der Lösung vom gewählten Gitter aus k-Punkten im Wetting-Layer genauer zu untersuchen, dabei könnte auch die Adaptation in k-Richtung oder eine Interpolation von weiteren k-Punkten betrachtet werden.

Innerhalb des numerischen Verfahrens gibt es beispielsweise bei der Berechnung der Koeffizienten in g_i in (6.33) Verbesserungsmöglichkeiten: Anstatt die Berechnung in der Matrix (6.26) in jeder Zeile mit einer Anlaufrechnung neu zu starten, könnte versucht werden, diese Matrix dreidimensional aufzustellen und so die wiederholte Integration der bis auf einen Faktor identischen Produkte zu vermeiden. Da diese Rechnungen allerdings nur mit reellen Zahlen stattfinden, wäre der Geschwindigkeitsgewinn nicht sehr groß.

Anhang

Folgende Parameter und Konstanten wurden für alle Rechnungen verwendet:

\hbar	0.6582 meVps	
π	3.141592653	
k_B	0.08617343 meV K ⁻¹	Boltzmann-Konstante
e_c	$1.602 \cdot 10^5$ pA ps	Elementarladung
m_{e0}	$5.686 \cdot 10^{-3}$ meV ps ² nm ⁻²	Freie Elektronenmasse
m_e	$0.067 \cdot m_{e0}$	Effektive Masse der Elektronen in InGaAs
m_h	$0.150 \cdot m_{e0}$	Effektive Masse der Löcher in InGaAs
ω_{LO}	$36/\hbar$	Frequenz der LO-Phononen
Γ_{LO}	$0.2 \cdot \hbar$	Dämpfungsrate der LO-Phononen
α	0.06	Fröhlich-Kopplungskonstante
ε_∞	12.01	Hochfrequenzdielektrizitätskonstante
L_{QD}	4.3 nm	Höhe des Quantenpunktes
L_{QW}	2.2 nm	Höhe des Wetting-Layers
ε_s^e	$-2.2 \cdot \hbar$	Energie der s-Schalen Elektronen
ε_s^h	$-1.1 \cdot \hbar$	Energie der s-Schalen Löcher
ε_p^e	$-0.8 \cdot \hbar$	Energie der p-Schalen Elektronen
ε_p^h	$-0.4 \cdot \hbar$	Energie der p-Schalen Löcher
E_{conf}^e	350 meV	z-Einsschlussenergie der Elektronen
E_{conf}^h	170 meV	z-Einsschlussenergie der Löcher
ρ_{QD}	10^{-10} cm ⁻²	Dichte der Quantenpunkte auf dem Wetting-Layer
ε_0	$1.41844 \cdot 10^6$ pA ² ps ² meV ⁻¹ nm ⁻¹	Elektronische Feldkonstante
β_{osc}^e	0.18 nm^{-1}	Inverse Oszillator Länge
β_{osc}^h	0.18 nm^{-1}	Inverse Oszillator Länge

(8.1)

Anleitung zur Verwendung des Programms

Der Vorliegende Solver löst Anfangswertprobleme mit Integro-Differentialgleichungen der Form

$$\begin{aligned} y &: \mathbb{R} \rightarrow \mathbb{C}^n \\ B &: \mathbb{R} \times \mathbb{C}^n \rightarrow \mathbb{C}^n \\ \frac{d}{dt}y(t) &= \int_0^t B(t-t', y(t-t')) \cdot y(t') dt' \\ y(0) &= 1. \end{aligned}$$

Der Solver ist im FORTRAN2003 Standard mit dem Intel-Compiler ifort Version 12.1.4 erstellt worden. Abweichungen davon können zu Problemem führen, neuere Intel-Compiler sollten funktionieren. Um den Solver einzusetzen, sind die folgenden Schritte notwendig:

Schritt 1: Definition der Objekte

Als erstes muss in `TOBJ_ops.f90` eine Struktur von $y(t)$ als `TOBJ%obj` festgelegt werden. Dabei sind Vektoren und Matrizen aber auch allgemeinere Objekte wie Sparse-Matrizen möglich. Die genauere Anzahl der Einträge in der Struktur der Objekte (`neqn`) wird später in Schritt zwei festgelegt. Für die gewählten Objekte müssen nun noch einige Operationen erklärt werden, wobei für Vektoren und Matrizen kann die Voreinstellung genutzt werden kann. Für allgemeinere Objekte müssen die folgenden algebraischen Operationen zwischen den Objekten und Skalaren zur Verfügung gestellt werden:

1. `TOBJ%init`
Festlegung der genaueren Größe eines Objektes durch `neqn` und initialisieren aller Einträge mit Null
2. `TOBJ%free`
Freigeben des Speichers, den ein Objekt belegt
3. `abs_TOBJ(TOBJ_in,TOBJ_inout)`
Der Betrag eines Objektes:
`TOBJ_inout = | TOBJ_in |`
4. `norm_TOBJ(TOBJ_in, real_inout)`
Die Norm eines Objektes:
`TOBJ_inout = || TOBJ_in ||`
5. `multScal2TOBJ(TOBJ_in,scal,TOBJ_inout)`
Elementweises Multiplizieren eines Objektes mit einem Skalar:
`TOBJ_inout = TOBJ_in .* scal`
6. `multTOBJ2TOBJp(TOBJ_in1, TOBJ_in2, p, TOBJ_inout)`
Elementweises Multiplizieren eines Objektes mit der Elementweisen Potenz p eines anderen Objektes:
`TOBJ_inout = TOBJ_in1 .* TOBJ_in2 .^ p`

7. `addTOBJ2sTOBJ(TOBJ_in1,scal,TOBJ_in2,TOBJ_inout)`
 Elementweises Addieren eines Objektes mit einem anderen Objekt, das zuvor elementweise mit einem Skalar multipliziert wurde:

$$\text{TOBJ_inout} = \text{TOBJ_in1} .+ \text{TOBJ_in2} .* \text{scal}$$
8. `addTOBJ2TOBJ(TOBJ_in1,TOBJ_in2,TOBJ_inout)`
 Elementweises Addieren eines Objektes mit einem anderen Objekt:

$$\text{TOBJ_inout} = \text{TOBJ_in1} .+ \text{TOBJ_in2}$$
9. `addscal2TOBJ(TOBJ_in,scal,TOBJ_inout)`
 Elementweises Addieren eines Skalars zu einem Objekt:

$$\text{TOBJ_inout} = \text{TOBJ_in} .+ \text{scal}$$
10. `summultTOBJ2TOBJ2mat(TOBJ_in1_vec,TOBJ_in2_vec,mat,idx_1_max,idx_2_max,TOBJ_inout)`
 Doppelsumme über Produkte von Einträgen von Vektoren von Objekten multipliziert mit jeweils einem anderen Skalar:

$$\text{TOBJ_inout} = \text{sum_2} \text{sum_1} \text{TOBJ_in1_vec}(\text{idx_1}) * \text{TOBJ_in2_vec}(\text{idx_2}) * \text{scal}(\text{idx_2},\text{idx_1})$$

 Diese Routine sollte aus Performancegründen so effizient wie irgend möglich programmiert werden, da der Solver etwa 80% der Zeit hiermit verbringt.

Schritt 2: Definieren von `get_B`

Bei der Initialisierung des ABM113 wird zu erst in `models.f90 model%init` ausgeführt. Hier müssen der Startwert der Differentialgleichung `model%Y_start` und die Anzahl der Gleichungen `neqn` festgelegt werden. In der procedure `save_to_file` kann definiert werden, welche Teile eines Objektes in eine Datei geschrieben werden, wenn in Schritt drei die entsprechenden Flags gesetzt werden. Die Standardeinstellung ist hier, dass alle Einträge geschrieben werden, dies kann für große Objekte allerdings zu Problemen führen, weshalb die Möglichkeit besteht, die zu schreibenden Daten einzuschränken.

Außerdem wird die procedure `model%get_B(d_time, y_d_time,dB)` benötigt, die die Auswertung $\text{dB} = \text{B}(\text{d_time}, \text{y_d_time})$ bereitstellt.

Schritt 3: Initialisieren

Die Initialisierung des ABM113 erfolgt mit `abm%init(t_start, t_end, abserr_abm, relerr_abm, maxorder, scale_factor, allow_overshoot, print_every, file_output_flag_y, file_output_flag_b, soln_size)`, wobei für die Argumente gilt:

- `t_start, t_end` (real): Start- und Endzeitpunkt der Differentialgleichung
- `abserr_abm, relerr_abm` (real): Geforderter absoluter und relativer Fehler
- `maxorder` (integer): Maximale Ordnung der im ABM113 verwendeten Verfahren (1-12 möglich, fast immer ist 12 sinnvoll)
- `scale_factor` (integer): Faktor, mit dem die Auflösung von `get_next_index_above` definiert wird, 100000 ist ein geeigneter Wert

- *allow_overshoot* (logical): Festlegung, ob der ABM113 über den Endzeitpunkt hinaus propagieren darf, im Normalfall `.true.`, nur bei Definitionslücken oder ähnlichem bei `t_end` `.false.` zu setzen
- *print_every* (integer): Festlegungen, bei jedem wie vielen erfolgreichen Schritt eine Ausgabe in der Konsole erfolgt
- *file_output_flag_y*, *file_output_flag_b* (logical): Festlegung, ob die Lösungen von y und B im Ordner Ergebnisse in Dateien geschrieben werden
- *soln_size* (integer): Anzahl der Punkte, auf denen die Lösung von y nach der Propagation interpoliert wird.

Schritt 4: Ausführen

Das Ausführen des ABM113 ist dann mit *abm%eval* möglich. Nach der erfolgreichen Propagation steht die Lösung von y auf dem mit *soln_size* geforderten, äquidistanten Gitter in *abm%soln* zur Verfügung.

Kontakt

Falls (in der Arbeit ungeklärte) Fragen bezüglich des Solvers oder der Programmstruktur auftreten oder eine Version des Solvers gewünscht wird, bitte E-Mail an *n.kemlein@gmail.com*.

Literatur

- [1] G. Czycholl. *Theoretische Festkörperphysik - Von den klassischen Modellen zu modernen Forschungsthemen*. Springer DE, Berlin, 3. aktualisierte Aufl. 2008 edition, 2007.
- [2] G. Engeln-Müllges. *Numerik-Algorithmen - Entscheidungshilfe zur Auswahl und Nutzung*. VDI-Verlag, Düsseldorf, 8. Aufl. edition, 1996.
- [3] H. Ibach and H. Lüth. *Festkörperphysik - Einführung in die Grundlagen*. Springer, Berlin, Heidelberg, 2009.
- [4] F. Jahnke. Greensfunction for the polaron problem. Manuskript.
- [5] F. Jahnke. Green'sche Funktionen und Relativistische Quantenmechanik. Vorlesungsmanuskript, 2011.
- [6] P. Linz. *Analytical and Numerical Methods for Volterra Equations*. SIAM, Philadelphia, 1985.
- [7] G. D. Mahan. *Many-Particle Physics*. Springer, Berlin, Heidelberg, 3rd ed. 2000 edition, 2000.
- [8] S. Sauvage. Long polaron lifetime in InAs/GaAs self-assembled quantum dots. *Phys. Rev. B*, 88, 2002.
- [9] K. Schuh. Polaronen in Halbleiter-Quantenpunkten, 2008.
- [10] W. Schäfer and M. Wegener. *Semiconductor Optics and Transport Phenomena*. Springer, Berlin, Heidelberg, 2002. Aufl. edition, 2002.
- [11] L. F. Shampine and M. K. Gordon. *Computer solution of ordinary differential equations*. W.H. Freeman & Co Ltd, Freiburg i.B., 1975.
- [12] A. Steinhoff. Mikroskopische Beschreibung des Ladungsträgereinfangs in Halbleiter-Heterostrukturen durch Elektron-Phonon-Wechselwirkung, 2011.
- [13] C. W. Überhuber and C. W. Ueberhuber. *Computer-Numerik 2*. Springer DE, Berlin, 1995.

Danke Alex! Für alles.