

# **Molekulardynamiksimulationen zur Untersuchung des Mischalkali–Effektes in silikatischen Gläsern**

DISSERTATION

Zur Erlangung des Grades

„Doktor der Naturwissenschaften“

am Fachbereich Physik

der Johannes Gutenberg–Universität in Mainz

vorgelegt von

Hans Rudolf Ribeiro Knoth

geboren in Rio de Janeiro, Brasilien

Mainz 2004





## Zusammenfassung

In der vorliegenden Arbeit wird mittels Molekulardynamik(MD)–Computersimulationen die Dynamik von verschiedenen Alkalisilikaten in der Schmelze und im Glas untersucht. Es ist bekannt, daß diese Systeme ionenleitend sind, was auf eine hohe Mobilität der Alkaliionen im Vergleich zu den glasbildenden Komponenten Si und O zurückzuführen ist. Im Mittelpunkt des Interesses steht der sog. Mischalkaliefekt (MAE), der in ternären Mischungen aus  $\text{SiO}_2$  mit zwei Alkalioxiden auftritt. Gegenüber Mischungen mit nur einer Alkaliionensorte weisen letztere Systeme eine signifikante Verlangsamung der Alkaliionendiffusion auf.

Zunächst werden zwei binäre Alkalisilikate simuliert, nämlich Lithiumdisilikat (LS2) und Kaliumdisilikat (KS2). Die Simulationen zeigen, daß der Ursprung der hohen Mobilität der Alkaliionen in der Struktur begründet ist. KS2 und LS2 weisen auf intermediären Längenskalen Ordnung auf, die in partiellen statischen Struktur Faktoren durch Prepeaks reflektiert ist. Die den Prepeaks zugrundeliegende Struktur erklärt sich durch perkolierende Netzwerke aus alkalioxidreichen Kanälen, die als Diffusionskanäle für die mobilen Alkaliionen fungieren. In diesen Kanälen bewegen sich die Ionen mittels Sprüngen (Hopping) zwischen ausgezeichneten Plätzen. In der Simulation beobachtet man für die hohen Temperaturen ( $4000 \text{ K} \geq 1500 \text{ K}$ ) eine ähnliche Aktivierungsenergie wie im Experiment. Im Experiment findet allerdings unterhalb von ca.  $1200 \text{ K}$  ein Crossover in ein Arrheniusverhalten mit höherer Aktivierungsenergie statt, welches von der Simulation nicht nachvollzogen wird. Das kann mit der in der Simulation nicht im Gleichgewicht befindlichen Si–O–Matrix erklärt werden, bei der Alterungseffekte beobachtet werden.

Am stärksten ist der MAE für eine Alkalikomponente, wenn deren Konzentrationsanteil in einem ternären Mischalkalisystem gegen 0 geht. Daher wird ein LS2–System untersucht, in dem ein Li–Ion gegen ein K–Ion getauscht wird. Der Einfluß des K–Ions ist sowohl lokal in den charakteristischen Abständen zu den ersten nächsten Nachbarn (NN) zu sehen, als auch in der ortsaufgelösten Koordinationszahlverteilung bis zu Längenskalen von ca.  $8.5 \text{ \AA}$ . Die Untersuchung der Dynamik des eingesetzten K–Ions zeigt, daß die Sprungwahrscheinlichkeit nicht mit der Lokalisierung, einem Maß für die Bewegung eines Teilchens um seine Ruheposition, korreliert ist, aber daß eine chemische Umgebung mit wenig Li– und vielen O–NN oder vielen Li– und wenig O–NN ein Sprungereignis begünstigt.

Zuletzt wird ein ternäres Alkalisilikat  $(0.5 \cdot \text{Li}_2\text{O})(0.5 \cdot \text{Li}_2\text{O})(2 \cdot \text{SiO}_2)$  (LKS2) untersucht, dessen Struktur alle charakteristischen Längenskalen von LS2 und KS2 aufweist. Es stellt sich also eine komplexe Struktur mit zwei perkolierenden Subnetzwerken für Alkaliionen ein. Die Untersuchung der Dynamik zeigt eine geringe Wahrscheinlichkeit dafür auf, daß Ionen in ein Subnetzwerk andersnamiger Ionen springen. Auch kann gezeigt werden, daß das Modellpotential den MAE reproduzieren kann, daß also die Diffusionskonstanten in LKS2 bei  $700 \text{ K}$  bis zu einer Größenordnung langsamer sind als in KS2 bzw. LS2. Der beobachtete Effekt stellt sich zudem vom funktionalen Verlauf her so dar, wie er beim MAE erwartet wird. Es wurde auch festgestellt, daß trotz der zeitlichen Verzögerung in den dynamischen Größen die Anzahl der Sprünge pro Zeit nicht geringer ist und daß für niedrige Temperaturen (d.h. im Glas) Sprünge auf den Nachbarplatz mit anschließendem Rücksprung auf die vorherige Position deutlich wahrscheinlicher sind als bei hohen Temperaturen (also in der Schmelze).

Die vorliegenden Resultate geben Aufschluß über die Details der Mechanismen mikroskopischer Ionenleitung in binären und ternären Alkalisilikaten sowie dem MAE.



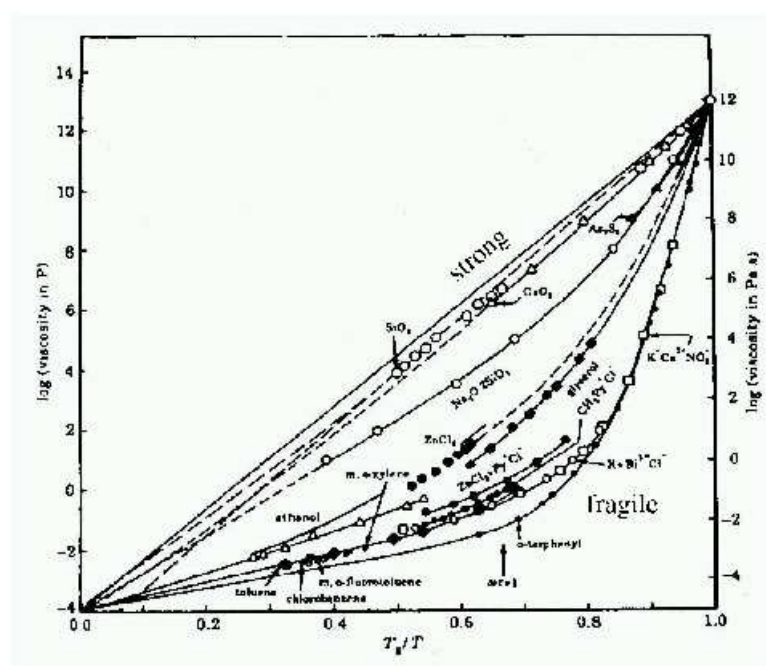
# Inhaltsverzeichnis

<b>1. Eine Einführung zu ionenleitenden Silikaten</b>	<b>1</b>
1.1. Einleitung zu ionenleitenden Gläsern . . . . .	1
1.2. Bisherige Simulationen ionenleitender Alkalisilikate . . . . .	4
1.3. Theorien zum MAE . . . . .	7
1.4. Modenkopplungstheorie und Ionenleiter . . . . .	9
<b>2. Simulationsdetails und Messgrößen</b>	<b>17</b>
2.1. Simulation: Durchführung und Methoden . . . . .	17
2.1.1. Molekular–Dynamik–Simulationen (MD) . . . . .	17
2.1.1.1. Der Velocity–Verlet Algorithmus . . . . .	18
2.1.1.2. MD Simulationen in verschiedenen Ensembles . . . . .	21
2.1.1.3. Anfangsbedingungen . . . . .	25
2.1.1.4. (Periodische) Randbedingungen . . . . .	26
2.1.1.5. Minimum Image Convention . . . . .	27
2.1.1.6. Nachbarlisten . . . . .	28
2.1.1.7. Ewald–Summation . . . . .	28
2.1.2. Mikroskopisches Modell . . . . .	33
2.1.2.1. Potential . . . . .	33
2.1.2.2. Vergleich mit dem Experiment . . . . .	36
2.1.3. Parallelisierung und praktische Durchführung . . . . .	37
2.1.4. Übersicht simulierter Systeme . . . . .	39
2.2. Analysemethoden . . . . .	41
2.2.1. Strukturelle Größen . . . . .	41
2.2.1.1. Die partielle Paarkorrelationsfunktion $g_{\alpha\beta}(r)$ . . . . .	41
2.2.1.2. Die Koordinationszahlverteilung . . . . .	42
2.2.1.3. Bond–Bond–Winkelverteilungen . . . . .	43
2.2.1.4. Ringlängenverteilungen . . . . .	43
2.2.1.5. Die partiellen statischen Strukturfaktoren $S_{\alpha\beta}(q)$ . . . . .	44
2.2.1.6. Der statische „Neutronenstreu“-Strukturfaktor $S_{\text{Neutron}}(q)$ . . . . .	45
2.2.1.7. Der elastische Strukturfaktor . . . . .	46
2.2.2. Dynamische Größen . . . . .	46
2.2.2.1. Mean Squared Displacement (MSD) . . . . .	46
2.2.2.2. Die van Hove Korrelationsfunktionen $G_s(r)$ und $G_d(r)$ . . . . .	47
2.2.2.3. (In)Kohärente intermediäre Streufunktionen $F_s(q, t)$ & $F(q, t)$ . . . . .	48

<b>3. Eigenschaften von Alkalidisilikaten</b>	<b>49</b>
3.1. Strukturelle Eigenschaften . . . . .	50
3.2. Dynamische Eigenschaften . . . . .	62
3.3. Alterungs-Phänomene . . . . .	69
<b>4. Binäres Lithiumdisilikat mit einem Fremdkaliumatom</b>	<b>81</b>
4.1. Strukturelle Besonderheiten um das Fremdion . . . . .	82
4.2. Dynamik des Fremdions . . . . .	88
<b>5. Ternäres Alkalidisilikat und der MAE</b>	<b>93</b>
5.1. Strukturelle Eigenschaften . . . . .	93
5.2. Dynamische Eigenschaften und der MAE . . . . .	105
5.2.1. Auftreten und Ausprägung des MAE . . . . .	105
5.2.2. Dynamik von ternärem $0.5(\text{Li}_2\text{O}) 0.5(\text{K}_2\text{O}) 2(\text{SiO}_2)$ . . . . .	110
<b>6. Zusammenfassung</b>	<b>121</b>
<b>A. Simulationsprogrammcode</b>	<b>125</b>
<b>Literaturverzeichnis</b>	<b>147</b>

# 1. Eine Einführung zu ionenleitenden Silikaten

## 1.1. Einleitung zu ionenleitenden Gläsern



**Abbildung 1.1.:** Veranschaulichung der Definition von starken (strong) und fragilen Glasbildnern. Die Glasübergangstemperatur  $T_g$  wird dabei via der Viskosität  $\eta(T_g) = 10^{13}$  Poise definiert.

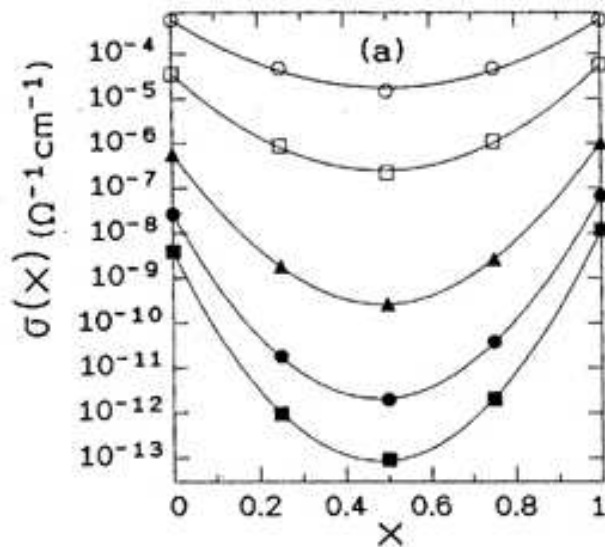
Die Dynamik von unterkühlten Flüssigkeiten und verwandte Phänomene des Glasübergangs sind seit langem ein Brennpunkt des Interesses [1, 2, 3, 4]. Der Grund dafür ist, daß für einen Glasbildner, der von seiner Schmelztemperatur zur Glasübergangstemperatur  $T_g$  heruntergekühlt wird, die Relaxationszeit um ungefähr 14 Größenordnungen langsamer wird und die Viskosität  $\eta$  dramatisch ansteigt, ohne eine signifikante Änderung seiner strukturellen Eigenschaften aufzuweisen. Die Untersuchung dieses Verhalten stellt für Experimentatoren und Theoretiker eine erhebliche Herausforderung dar.

In Abbildung 1.1 sieht man eine Auftragung des Logarithmus der Viskosität als Funktion von  $T_g/T$ , wobei  $T_g$  nach einer Definition von Laughlin und Uhlman willkürlich über den Wert



### Eine kurze Geschichte des Glases (vgl. [5]).

5000 v.Chr.	<p><b>Die Entdeckung von Glas:</b> Glas existiert seit dem Erkalten der Erdkruste. Es entsteht wenn bestimmte Arten Gestein durch hohe Temperaturen schmelzen, z.B. Vulkaneruptionen, Blitz- oder Meteoriteneinschläge, und dann schnell erkalten. Schon der Steinzeitmensch soll Schneidewerkzeuge aus Obsidian (Glas vulkanischer Herkunft) benutzt haben.</p> <p>Nach dem frühen römischen Historiker Plinius (AD 23–79) „entdeckten“ Phönizische Händler die Existenz von Glas in der Region von Syrien per Zufall: Sie legten Kochtöpfe auf Nitratablöcken nahe dem Kochfeuer ab. Diese Blöcke schmolzen in der intensiven Hitze, vermischten sich mit dem Sand des Strandes an dem sie lagerten, und ergaben eine durchsichtige Flüssigkeit.</p>
3500 v.Chr.	<p>Das älteste absichtlich hergestellte Glas, hauptsächlich undurchsichtige Glasperlen, findet sich in Mesopotamien und Ägypten. Im 3. Jahrtausend v. Chr. wurden die grundsätzlichen Rohmaterialien von Glas zur Glasur von Keramik verwendet.</p>
1500 v.Chr.	<p>Glasvasenfragmente in Mesopotamien (ältester Hinweis auf Hohlglasproduktion). Auch in Ägypten wird eine Hohlglasproduktionsmethode entwickelt. Frühe Glasproduktionsaktivitäten auch in Griechenland, China und Nordtirol.</p>
9. Jhd. v.Chr.	<p>Wieder-erstarken der Glasproduktion in Mesopotamien. In den folgenden 500 Jahren verlagert sich diese nach Alexandria, von wo sie nach Italien gelangt.</p>
650 v.Chr.	<p>Die erste „<b>Glasproduktionsanleitung</b>“ findet sich in der Bibliothek des Assyrischen Königs Ashurbanipal.</p>
27 v.Chr.-14 n.Chr.	<p>Großer Durchbruch in der Glasproduktionstechnik in der Sidon-Babylon Gegend: geblasenes Glas mittels eines hohlen Metallstabs.</p>
100 n.Chr.	<p>Die Römer verbreiteten in ihrem Reich Glas und Glasproduktion. Auch die ersten <b>Glasfenster</b> gehen auf das römische Reich zurück.</p>
1000 n.Chr.	<p>Eine tiefgreifende Änderung der Glasproduktion bahnt sich an. Da Natron schwer zu importieren war, wurde es schrittweise durch Pottasche (Kalisalze) von verbrannten Bäumen ersetzt. Ab diesem Zeitpunkt unterschied sich Glas nördlich der Alpen von dem z.B. in Italien, dessen Hauptbestandteil Natron war.</p>
15.-17. Jhd. 1674 n.Chr.	<p>In Murano wird <b>Kristallglas</b> aus Quarzsand und Pottasche hergestellt.</p> <p><b>Bleikristall.</b> Der Englische Glasmacher G. Ravenscroft (1618-1681) patentierte sein neues Glas 1674. Er sollte einen Ersatz für das Kristallglas aus Murano entwickeln. Durch die Verwendung von hohen Anteilen an Bleioxyd statt Pottasche erhielt er ein Glas mit hohem Brechungsindex.</p>
spätes 19. Jhd.	<p>Einer der Urväter der modernen Glasforschung war der deutsche Forscher <b>Otto Schott</b> (1851-1935), der wissenschaftliche Methoden zur Untersuchung der Effekte verschiedener Elemente im Glas, auf optische und thermische Eigenschaften hin verwendete. Zusammen mit <b>Ernst Abbe</b> (1840-1905) untersuchte er optische Gläser. Ein weiterer Beitrag zur Massenproduktion kam von Friedrich Siemens, der Durchlaufhochöfen einführte, mittels derer größere Mengen geschmolzenes Glas produziert werden konnte.</p>



**Abbildung 1.2.:** Beispiel für den MAE in einem Borosilikat aus [7]. Die Zusammensetzung ist  $x(\text{Na}_2\text{O}) - (1 - x)(\text{Li}_2\text{O}) - 3(\text{B}_2\text{O}_3)$ , die Temperaturen reichen von 600 K (oberste Kurve) bis 325 K (unterste Kurve). Wie gut sichtbar ist, ist der MAE am stärksten für die niedrigste Temperatur, wo er über mehrere Größenordnungen geht. Das Minimum der Leitfähigkeit  $\sigma(x)$  findet sich hier bei  $x = 0.5$ .

$\eta(T_g) = 10^{13}$  Poise festgelegt ist. Die Auftragung in Abbildung 1.1 suggeriert, daß die Glasbildner grob in zwei Klassen eingeteilt werden können: Einmal die der „starken“ Glasbildner (z.B.  $\text{SiO}_2$ ), die über den gesamten Temperaturbereich ein Arrheniusverhalten aufweisen und „fragile“ Glasbildner, die ein stark temperaturabhängiges Verhalten aufweisen. In dieser Abbildung ist auch das Viskositätsverhalten für Natriumdisilikat aufgetragen. Im Vergleich zu reinem amorphem  $\text{SiO}_2$  sieht man, daß die Zugabe von Natriumoxid zu  $\text{SiO}_2$  zu einer Schwächung der „starken“ Glaseigenschaft geführt hat, Natriumdisilikat aber immer noch eher den starken Glasbildnern zuzuordnen ist.

Der Glasbildner  $\text{SiO}_2$  bildet ein ungeordnetes Netzwerk aus, in dem  $\text{SiO}_4$ -Tetraeder über die Sauerstoffatome an den Ecken miteinander verknüpft sind. In Mischungen von  $\text{SiO}_2$  mit einem Alkalioxid  $\text{X}_2\text{O}$  (wobei X eine Alkaliionensorte ist) sind die physikalischen Eigenschaften sehr verschieden von denen in  $\text{SiO}_2$ . Das hat seinen Grund darin, daß Alkaliionen sogenannte Netzwerkmodifikatoren sind, die die Vernetzung zwischen den  $\text{SiO}_4$ -Tetraedern aufbrechen. Charakteristisch für Alkalisilikate ist, daß sie Strukturbildung auf Längenskalen, die deutlich größer als die Abstände benachbarter Atome sind, zeigen. Aber nicht nur die Struktur wird durch die Zugabe von Alkaliionen modifiziert, sondern es tritt auch eine Zeitskalentrennung der Netzwerkcomponenten (Si und O) und den Alkaliionen ein: Mobile Alkaliionen bewegen sich in einer vergleichsweise immobilen Si–O–Matrix [6].

Tauscht man in einem binären Alkalisilikatglas einen Teil der Alkaliionen durch die einer anderen Atomsorte aus, z.B. indem man in einem Kaliumsilikat in der Schmelze die Kaliumionen teils durch Lithiumionen ersetzt und die Schmelze abkühlt, so stellt man eine Anomalität in den Transporteigenschaften wie z.B. der elektrischen Leitfähigkeit und der Selbstdiffusion, fest. Diese Anomalität, daß die genannten Größen in Mischalkalisystemen deutlich kleiner sind, als in binären Alkalisilikaten, ist um einige Größenordnungen größer als eine lineare Änderung mit der Zusammensetzung es vermuten lassen würde. Der Effekt ist dabei umso ausgeprägter, je tiefer die Temperatur ist (siehe Abbildung 1.2). Diese Erscheinung wird *Mixed Alkali Effect* (MAE) genannt.

### Die Entdeckung des Mixed Alkali Effect [8]

Bis vor 1885 verwendete man für die meisten physikalischen Temperaturmessungen Glasthermometer. Diese waren jedoch damals mit einem inakzeptabel großen Messfehler belastet. Wurde ein Glasthermometer zur Messung höherer Temperaturen verwendet, so kehrte der Quecksilberfaden nach der Messung nicht mehr an den alten Ausgangspunkt zurück. Grund dafür war die Erweiterung der Glaskugel die das Quecksilber enthält. Der Fehler betrug bei einer Erhitzung des Thermometers auf 100°C und anschließender Abkühlung auf 0°C mittels Eiswasser  $-0.7^{\circ}\text{C}$ . Interessanterweise ließ dieser Fehler nach genügend langer Lagerung nach.

Nach einigen Vorarbeiten von Weber, Schott und Wiebe erkannte Otto Schott, daß der Effekt der Nullpunktsdepression mit dem gleichzeitigen Vorhandensein von Natrium und Kalium im alten Thermometerglas zusammenhing. Man hatte damit eine makroskopische Manifestation des Mixed Alkali Effect entdeckt.

Daraufhin entwickelte er das sogenannte *Jenaer Normalglas 16<sup>III</sup>* sowie das *Borosilicat-Thermometerglas 59<sup>III</sup>*, die beide eine deutlich geringere Nullpunktsdepression aufweisen.

Rein empirisch sind Methoden zur Beherrschung des MAE schon seit ca. 100 Jahren bekannt (vgl. Infokasten auf Seite 4). Dennoch sind in Hinsicht auf Struktur und Dynamik noch viele Fragen offen, weshalb der MAE in den letzten Jahren einen Forschungsschwerpunkt darstellt, dem vor allem mittels Experimenten und Computersimulationen nachgegangen wird (vgl. Kapitel 1.2). Gerade diese zwei Disziplinen sind in der Lage, sich über Messungen und Vorhersagen gegenseitig zu unterstützen, um schlussendlich eine valide mikroskopische Beschreibung des MAE zu finden (vgl. auch Kapitel 2.2.1.5).

Schwerpunkt dieser Arbeit soll die mikroskopische Untersuchung des Mischalkaliefekts in sogenannten ternären Alkalisilikaten sein. Ternäre Alkalisilikate sind Silikate mit zwei verschiedenen Alkalioxiden, in dieser Arbeit  $\text{Li}_2\text{O}$  und  $\text{K}_2\text{O}$ . Dazu werden zunächst einmal in Kapitel 3 die Eigenschaften der binären Alkalisilikate  $(\text{Li}_2\text{O})(2\text{SiO}_2)$  und  $(\text{K}_2\text{O})(2\text{SiO}_2)$  auf Gemeinsamkeiten und Unterschiede hin untersucht. Daran schließt sich in Kapitel 4 die Untersuchung eines binären Alkalisilikats mit einem Fremdion an, um die Reichweite der Auswirkungen der Zugabe eines fremdnamigen Alkaliions auf Struktur und Dynamik zu ergründen. Mit diesen Ergebnissen im Hinterkopf soll dann in Kapitel 5 der Mischalkaliefekt genauer betrachtet werden, indem ein ternäres Alkalisilikat analysiert wird, welches gleichberechtigt Lithium- und Kaliumoxid enthält.

## 1.2. Bisherige Simulationen ionenleitender Alkalisilikate

Untersuchungen von Alkalisilikaten mittels Computersimulationen haben sich meist mit der Frage befasst, wie die hohe Mobilität der Alkaliionen zu erklären ist. Die grundsätzliche Idee, daß die hohe Mobilität von Alkaliionen in Alkalisilikaten (und auch anderen Systemen) durch eine Bewegung der Alkaliionen durch ausgezeichnete Pfade („*preferential pathways*“) zu erklären ist, stammt ursprünglich von Greaves (z.B. [9, 10, 11, 12]). In vielen Arbeiten wurde diese Idee aufgegriffen und weiter verfeinert, u.a. von Sanz [13], Smith [10, 11, 12] und vielen

anderen. Hier sollen nun einige Beispiele bisheriger Simulationen zu ionenleitenden Silikaten vorgestellt werden, um zu vermitteln, welche Ansätze genau bisher dabei verfolgt wurden. Dabei werden Beispiele zu binären Silikaten<sup>1</sup> vorgestellt und auch Arbeiten zur Untersuchung des Mischalkaliefekts (MAE).

Bei der Untersuchung von Natriumsilikaten, speziell zu Natriumdi- (NS2) und Natriumtrisilikat (NS3) (aber auch zu anderen Zusammensetzungen bis zu NS20) gibt es einige Arbeiten auf Basis von Molekulardynamiksimulation mit dem sogenannten BKS-Potential (vgl. dazu Infokasten auf Seite 34) [6, 14, 15, 16, 17]. Im Zentrum dieser Arbeiten liegt die Aufklärung des Zusammenhangs zwischen der mikroskopischen Struktur und der hohen Mobilität der Alkaliionen. Als typisch für die Struktur von Natriumsilikaten stellt sich ein Prepeak in partiellen Strukturfaktoren (vgl. Kapitel 2.2.1.5) bei  $0.95 \text{ \AA}^{-1}$  heraus. Die damit zusammenhängende Struktur ist ein perkolierendes Netzwerk aus natriumreichen Kanälen in der Matrix des tetraedrischen Si-O-Netzwerks. Diese Kanäle bleiben auf der Zeitskala der Matrix erhalten, wobei die Na-Ionen sich auf einer schnelleren Zeitskala diffusiv in diesen Kanälen fortbewegen (Zeitskalentrennung zwischen Matrix und Natriumionen).

Auch auf dem BKS-Potential sind Arbeiten von Jund et al. aufgebaut [18, 19, 20, 21]. In diesen geht es auch um die Frage nach Kanälen für die Alkaliionendiffusion. Die Problemstellung, die dabei aber untersucht wird, ist eine andere. Jund et al. betrachten „dynamische Diffusionskanäle“. Diese definieren sich durch den mobilsten Anteil der Alkaliionen (konkret: die mobilsten 5% der Ionen). Von diesen Ionen werden die Trajektorien verfolgt, bevorzugte Plätze lokalisiert und Korrelationen zwischen diesen erkannt.

Zu verschiedenen Lithiumsilikaten gibt es Simulationen von Banhatti and Heuer [22], in denen das gleiche Potential verwendet wurde, wie in dieser Arbeit (vgl. Kapitel 2.1.2.1, allerdings wurde nicht die Korrektur verwendet, die im Infokasten auf Seite 35 eingeführt wird). In diesen Arbeiten wurde einmal die Güte des Potentials untersucht, indem man strukturelle (z.B. Nächste-Nachbar-Abstände) und dynamische Größen (z.B. aus den Diffusionskonstanten berechnete Aktivierungsenergien) mit experimentellen Daten verglichen hat. Auch wurde der Zusammenhang zwischen statischen und dynamischen Eigenschaften in Abhängigkeit zur  $\text{Li}_2\text{O}$ -Konzentration untersucht, z.B. wie die Aktivierungsenergie mit steigender  $\text{Li}_2\text{O}$ -Konzentration abnimmt. Anschließende Arbeiten von Lammert et al. [23], Lammert and Heuer [24] befassen sich alleine mit Lithiummetasilikat  $(\text{Li}_2\text{O})(\text{SiO}_2)$ , also einem System mit relativ hoher  $\text{Li}_2\text{O}$ -Konzentration.

Die Simulationen von Heuer et al. werden im  $NVT$ -Ensemble durchgeführt. Die Dichte wurde also über den gesamten untersuchten Temperaturbereich konstant gehalten, z.B.  $2.34 \text{ g/cm}^3$  für  $(\text{Li}_2\text{O})(\text{SiO}_2)$ . Ermittelt wurden die Dichte für verschiedene Zusammensetzungen aufgrund von Experimenten und einer daraus ermittelten empirischen Dichteformel von Doweidar [25] bei Normalbedingungen, also bei Raumtemperatur. Simulationen im  $NVT$ -Ensemble haben den Nachteil, daß sie einen direkten Vergleich mit dem Experiment erschweren, wo ja meist im  $NpT$ -Ensemble gearbeitet wird. Das ist bei der Simulation von Alkalisilikaten von Bedeutung, da diese Systeme im  $NVT$ -Ensemble einen Druck aufweisen, der stark temperaturabhängig ist. Bei der bei Raumtemperatur ermittelten experimentellen Dichte hat man

---

<sup>1</sup>Binär heißt in diesem Zusammenhang, daß zwei Sorten Oxide im Glas sind. Ternäre Gläser enthalten entsprechend drei Sorten Oxide.

daher bei hohen Temperaturen ( $T > 2000$  K) hohe Drücke von mehreren GPa. Um einen besseren Vergleich mit dem Experiment zu ermöglichen, wurden daher in der vorliegenden Arbeit Simulationen bei konstantem Druck ( $p = 0$ ) durchgeführt.

Zum Mischalkaliefekt (MAE) gibt es interessante Arbeiten von Balasubramanian and Rao [26, 27]. Diese basieren auch auf klassischen Molekulardynamiksimulationen, allerdings wurde dort das Born–Mayer–Huggins–Paarpotential verwendet, mit Parametern von Tesar and Varshneya [28]. Simuliert wurde Alkalisilikat mit 35 mol% Alkalianteil, wobei dieser Alkalianteil durch verschiedene Anteilsverhältnisse von Kalium- und Lithiumionen zusammengesetzt war. Im wesentlichen wurden zur Auswertung der Dynamik die van–Hove–Korrelationsfunktionen<sup>2</sup> verwendet. Mittels diesen konnte man zum einen feststellen, daß sich die Ionen bei niedriger Temperatur (1000 K) durch Sprünge, das sogenannte „hopping“, in die Position ihrer nächsten Nachbarn fortbewegen. Daraus kann man folgern, daß es ausgezeichnete Ionenplätze in der Si–O–Matrix gibt. Für diese Plätze stellt man auch verschiedene chemische Ordnungen fest, d.h. an diesen Plätzen sind die Ionen je nach Sorte mit einer anderen Anzahl von Sauerstoffen koordiniert. Desweiteren ergab sich mittels des „distinct“-Anteils der van–Hove–Korrelationsfunktion, daß die Frage, ob ein Kaliumion in einen ausgezeichneten Lithiumplatz (oder andersherum) springt, nicht von der Größe der Ionen abhängt (was man vermuten könnte, da ein großes Ion einfach nicht in den Platz reinpasst, in dem sich vorher ein kleines Ion aufgehalten hat), sondern von der chemischen Umgebung, die sich durch die vorherige Anwesenheit des Ions an dem Platz eingestellt hat. Man spricht in diesem Fall von einer Art „Gedächtniseffekt“.

Einen etwas anderen Ansatz als die bisherigen Arbeiten verfolgen Swenson and Adams [29]. Sie verwenden die *Reverse–Monte–Carlo*-Methode (RMC) auf Grundlage von Röntgen- und Neutronenstreudaten. Vereinfacht gesagt ist die RMC eine Methode, Teilchenkonfigurationen auszuwürfeln, die möglichst gut zu experimentell gewonnenen Daten (z.B. Struktur Faktoren) passen. Das große Problem dieser Methode ist jedoch, daß die Zuordnung von Teilchenkonfigurationen zu experimentellen Daten nicht eindeutig ist. So hängt die Konfiguration, die man aus der RMC–Prozedur am Ende erhält, stark von der Anfangskonfiguration ab. In der Arbeit [29] erhalten Swenson and Adams bevorzugte Ionenpfade, allerdings begründen sie das Abfallen der Ionenleitfähigkeit im Mischsystem durch Blockadeeffekte, d.h., daß ein Ion oder mehrere Ionen der Sorte B in bevorzugten Plätzen der Sorte A sitzen und so den Diffusionspfad blockieren. Das ist ein Ergebnis, welches nicht im Einklang mit den Ergebnissen dieser Arbeit ist. Man muß dabei aber auch bedenken, daß, wie bereits gesagt, das „Auswürfeln“ von Teilchenkonfigurationen in der RMC eine gewisse Beliebigkeit hat und zudem nur strukturelle Informationen liefert, die direkt eigentlich keine Aussage über die Dynamik zulassen.

Als letztes sei hier noch eine Arbeit von Habasaki and Hiwatari [30] diskutiert, da darin das gleiche Potential und die gleichen Ionensorten verwendet wurden, wie in dieser Arbeit. Simuliert wurden zwei Metasilikate,  $\text{Li}_2\text{SiO}_3$  und  $\text{LiKSiO}_3$ . Hier sollen nicht die Ergebnisse dieser Arbeit besprochen werden, da sie offensichtlich auf fehlerhaften Auswertungen basieren.<sup>3</sup> Die in dem Artikel aufgeführten mittleren Verschiebungswerte (vgl. Kapitel 2.2.2.1) zeigen im diffusiven Regime für Zeiten ab ca. 100 ps ein erratic Verhalten, teils flachen die Kurvenverläufe einfach ab, teils fallen sie, teils gibt es Zwischenplateaus. Dieses Verhalten zeigt sich

---

<sup>2</sup>vgl. Kapitel 2.2.2.2.

<sup>3</sup>Es wird vermutet, daß bei der Auswertung der Daten das Berücksichtigen der periodischen Randbedingungen vergessen wurde.

in beiden untersuchten Systemen. Vergleiche mit anderen Arbeiten zu den gleichen Systemen mit dem gleichen Potential [22, 24] zeigen, daß dieses Verhalten auf Fehler in der Simulation oder der Auswertung zurückzuführen ist.

Weitere Arbeiten, auf die hier nicht im einzelnen eingegangen werden soll, die sich aber auch mit Ionenleitung (teils auch mit dem MAE) befassen, finden sich in den Referenzen [28, 31, 32, 33, 34].

### 1.3. Theorien zum MAE

Zunächst einmal stellt sich der MAE als uninteressant für technologische Anwendungen dar. Im speziellen Fall unter anderem auch, da Alkalisilikate bei Kontakt mit Feuchtigkeit sofort beginnen zu schäumen [8]. Dieser Effekt erschwert experimentelle Messungen nicht unerheblich [35]. Das Verständnis des MAE ist jedoch von eminenter Wichtigkeit zum Verständnis der Ionentransportmechanismen in Gläsern mit mobiler Alkalikomponente.

Im folgenden soll ein kleiner Überblick über die existierenden Theorien zum MAE gegeben werden. Die erste und sehr verblüffende Erklärung des MAE wurde von Myuller [36] vorgeschlagen. Er ging davon aus, daß die Wechselwirkung zwischen der Glasmatrix und den Ionen nicht durch die Anwesenheit andersnamiger Ionen beeinflusst wird. Die Aktivierungsenergie der einzelnen Ionen steigt dann nur deshalb an, da die partielle Konzentration einer Sorte Ionen im System genauso gering ist, wie die in einem binären Silikatglas ohne die Anwesenheit der anderen Sorte Ionen. Nichtsdestotrotz gibt es keine physikalische Begründung für Myuller's Modell, da die zweite Sorte Ionen sowohl nichtbrückenbildende Sauerstoffe einführt, die mit allen Ionen wechselwirken, als auch die Maschen der Glasmatrix aufgrund ihrer anderen Größe anders deformieren.

Die Annahme, daß der MAE etwas mit einer Potentiallandschaft zu tun hat, wurde bereits früh von Stevels [37] eingeführt. Die weitere Annahme, daß kleinere Ionen in Konkurrenz zu größeren Ionen stehen, um die Minima der Potentiallandschaft zu besetzen, wird durch Molekulardynamiksimulationen [38] unterstützt.

Ein relativ neues Modell, das sogenannte *dynamische Strukturmodell*, basiert auf zunächst unabhängigen Arbeiten [39, 40, 41, 42, 43] und einer späteren, gemeinsamen Arbeit [44] von Bunde, Ingram, Funke und Mitarbeitern (BFI-Modell [45]). Grundlage für die Überlegungen, die zum BFI-Modell führten, lieferten EXAFS<sup>4</sup> Experimente [47, 48]. In diesem Modell

---

<sup>4</sup>*Extended X-ray Absorption Fine Structure* [46] wird durchgeführt, indem man einen monochromatischen Röntgenstrahl auf die Probe richtet. Die Photonenenergie wird langsam erhöht, so daß es eine der Absorptionskanten der Elemente in der Probe kreuzt. Unterhalb der Absorptionskante können die Photonen die Elektronen nicht relevant anregen, daher ist die Absorption niedrig. Wenn die Photonenenergie jedoch gerade ausreicht, um die Elektronen anzuregen, steigt die Absorption rapide an (Absorptionskante). Die resultierenden Photoelektronen haben eine niedrige kinetische Energie und können durch die das emittierende Atom umgebenden Atome zurückgestreut werden. Diese Streuwahrscheinlichkeit hängt von der Energie der Photoelektronen ab.

Diese Rückstreuung von Photoelektronen beeinflusst, ob die Röntgenphotonen überhaupt absorbiert werden, da die Wellenfunktion des Photoelektrons von den umliegenden Atomen gestreut wird. Das bedeutet, der Endzustand  $\langle F \rangle$  des Photoelektrons ist abhängig von der Interferenz der Umgebung. Damit ist das Matrixelement, das die Übergangswahrscheinlichkeit zwischen Ausgangszustand  $\langle I \rangle$  des Photoelektrons und dem

wird wie folgt eine Interaktion zwischen den Ionen vorgeschlagen: Nach einer gewissen Zeit  $\tau$ , nach der ein  $A$ -Ion in einen Ionenplatz  $C$  eingetreten ist, verwandelt sich dieser Platz in einen  $\tilde{A}$ -Platz, einen Platz dessen lokale Umgebung auf die Nachbarschaft mit einem  $A$ -Ion eingestellt ist. Wenn sich das Ion weiterbewegt, verbleibt die „Erinnerung“ an das  $A$ -Ion und ein leerer  $\tilde{A}$ -Platz verbleibt für eine gewisse Zeit. So wird die Potentiallandschaft am Ort  $C$  eine Funktion der Zeit, da strukturelle Umordnungen im Glasnetzwerk oder in den umgebenden Ionen langsam im Vergleich zu den Sprungbewegungen (*Hopping*) der Ionen sind. Sprünge der Ionen in  $\tilde{A}$ -Plätze sind effektiver für langreichweitige Diffusion, da Sprünge in energetisch ungünstigere  $C$ -Plätze eine höhere Wahrscheinlichkeit eines Rücksprungs in die Position vor dem Sprung nach  $C$  bedeuten. So wird eine geringere Aktivierungsenergie bei höherem Alkalianteil erklärt. Die gleichen Annahmen zum Verhalten sollen nun für ein anderes, größeres Ion der Sorte  $R$  gelten. Um den Mischalkaliefekt zu erklären, muß noch die zusätzliche und äußerst wichtige Prämisse gemacht werden, daß ein  $A$ -Ion nicht bevorzugt in einen  $\tilde{R}$ -Platz springt und vice versa für die  $R$ -Ionen. Das ist aber für die  $A$ -Ionen schwer zu verstehen. Es ist klar, daß es für das größere  $R$ -Ion energetisch ungünstig ist, in einen kleinen  $\tilde{A}$ -Platz zu springen, aber  $A$ -Ionen sollten den geräumigeren  $\tilde{R}$ -Plätzen den engeren  $\tilde{A}$ -Plätzen den Vorzug geben. Um mit den Annahmen des BFI-Modells (für eine Ionensorte) vereinbar zu bleiben, muß vorausgesetzt werden, daß die Veränderung der Ladungen um einen Ionenplatz die strukturellen Änderungen überkompensiert. Zusammengefasst heißt das, wenn die nötigen Voraussetzungen, daß ein  $A$ -Ion einem  $\tilde{A}$ - oder  $C$ -Platz den Vorzug vor einem großen  $\tilde{R}$ -Platz gibt (und vice versa), fehlen, bricht die Erklärung des Mischalkaliefektes zusammen. Sind die Zeiten für die Relaxation der  $\tilde{A}$ - und  $\tilde{R}$ -Plätze in  $C$ -Plätze groß im Vergleich zur Wartezeit, bis ein weiteres Ion den Platz besucht, degeneriert das BFI-Modell in ein Myuller-Modell, in dem  $A$ -Ionen nur über  $\tilde{A}$ - oder  $C$ -Plätze migrieren (und entsprechend für  $R$ -Ionen).

Die daraus resultierenden Daten geben zwar die Richtung der Änderung der Aktivierungsenergien richtig wieder, nicht aber deren Abhängigkeit mit dem Konzentrationsverhältnis. Als Konsequenz stimmt die Krümmung der Aktivierungsenergien als Funktion der Alkaliionenzusammensetzung nicht mit den experimentellen Ergebnissen überein. Die richtige Krümmung erklären zu können, ist jedoch sehr wichtig. Sonst könnte der MAE einfach mittels der Veränderung der Aktivierungsenergie in Abhängigkeit von der Konzentration der einzelnen Ionensorten in binären Gläsern erklärt werden.

Aber es wird durchaus weiter am dynamischen Strukturmodell gearbeitet. In einer Arbeit von Maass [43] wird das Modell um Coulomb-Wechselwirkungen erweitert. In diesem Modell stimmt dann auch die Krümmung.

Ein weiterer Ansatz von Kirchheim [45] besagt, daß drei Faktoren die Mobilität von Ionen beeinflussen: Die Coulomb-Wechselwirkung zwischen den Alkaliionen und den mit ihnen koordinierten Sauerstoffen, eine breite Verteilung von Energien auf Ionenplätzen, die aus struktureller bzw. Coulombscher Unordnung herrührt und die Packungsdichte des Netzwerks. Es wird angenommen, daß in ternären Gläsern die Alkaliionen in Konkurrenz um die Plätze mit

---

Endzustand  $\langle F \rangle$  beschreibt, direkt abhängig von den umgebenden Atomen.

Das Gesamtergebnis ist eine Oszillation auf der Seite der hohen Energien der Absorptionskante. Aus diesen kann man die Atomnummern, Abstand und Koordination der umgebenden Atome untersuchen.

Zur Variation der Photonenenergie verwendet man Synchrotronstrahlung.

den niedrigsten Energien stehen. Unter der Annahme, daß das kleinere Alkaliion diesen Konkurrenzkampf gewinnt, da es näher an die Sauerstoffe herankommt, steigt die Aktivierungsenergie der kleineren Alkaliionensorte durch die Zugabe einer anderen, größeren Alkaliionensorte. Die Aktivierungsenergie für die größere Alkaliionensorte steigt aber auch, da sie durch ein Netzwerk wandern müssen, dessen Maschengröße durch die Präsenz von kleineren Alkaliionen verringert wurde.

## 1.4. Modenkopplungstheorie und Ionenleiter

Die Dynamik unterkühlter Flüssigkeiten und die verwandten Phänomene des Glasübergangs sind schon lange ein Forschungsschwerpunkt, nicht zuletzt, weil die Relaxationszeit sich vom Zustand einer „normalen“ Flüssigkeit hinunter in die typische Relaxationszeit von der Größenordnung 1 ps um ca. 14 Größenordnungen verlangsamt, ohne signifikante Änderung der strukturellen Eigenschaften. Trotz intensiver Untersuchung, sowohl experimentell als auch theoretisch, sind die grundlegenden Mechanismen nicht verstanden. Die einzige Theorie bisher, die viele Vorhersagen macht, die experimentell oder über Computersimulationen getestet werden können, ist die *Moden-Kopplungs-Theorie* (MCT) [49].

Konkret<sup>5</sup>: Die Dynamik einer Flüssigkeit wird oft über die Dichteautokorrelationsfunktion

$$F(\vec{q}, t) = \frac{1}{N} \langle \rho^*(\vec{q}, t) \rho(\vec{q}, 0) \rangle \quad \text{mit} \quad \rho(\vec{q}, t) = \sum_{j=1}^N \exp(i\vec{q} \cdot \vec{r}_j(t)), \quad (1.1)$$

oft auch intermediäre Streufunktion genannt, beschrieben (vgl. auch Kapitel 2.2.2.3), mit dem Wellenvektor  $\vec{q}$ , der Teilchenzahl  $N$  und der Position  $\vec{r}_j(t)$  des Teilchen  $j$  zur Zeit  $t$ .  $F(\vec{q}, t)$  ist von besonderem Interesse, da diese Größe sowohl in Streuexperimenten als auch in Computersimulation gemessen werden kann.

Für ein isotropes System kann die Bewegungsgleichung für  $F(\vec{q}, t)$ , die *Mori-Zwanzig-Gleichung*, geschrieben werden als

$$\ddot{F}(q, t) + \Omega^2(q)F(q, t) + \underbrace{\int_0^t [M^0(q, t-t') + \Omega^2(q)m(q, t-t')] \dot{F}(q, t') dt'}_{\text{Memorykern}} = 0, \quad (1.2)$$

mit der mikroskopischen Frequenz  $\Omega$ , die aus dem statischen Strukturfaktor  $S(q)$  mittels

$$\Omega^2(q) = \frac{q^2 k_B T}{m S(q)} \quad (1.3)$$

berechnet werden kann (mit  $m$  der Masse und  $k_B$  der Boltzmannkonstante (siehe Tabelle 2.1)). Der *Memorykern* besteht aus zwei Anteilen:

- $M^0(q, t)$  beschreibt die Kurzzeitdynamik.

<sup>5</sup>Die Argumentation wird hierbei im wesentlichen aus [50, 51] übernommen.



- $m(q, t)$  wird relevant für stark unterkühlte Systeme.

$M^0(q, t)$  liefert, anders als im Fall von normalen Flüssigkeiten in der Nähe des Tripelpunktes, keinen wichtigen Beitrag zur Glasdynamik. Daher kann man unter der Modellannahme, daß  $M^0(q, t)$  nur bei  $t = 0$  einen großen Anteil hat, für die Kurzzeitdynamik eine Näherung durch weißes Rauschen mittels  $\delta$ -Funktion machen, so daß

$$M^0(q, t) = \nu(q)\delta(t). \quad (1.4)$$

Damit wird die Gleichung (1.2) zu der eines gedämpften harmonischen Oszillator mit einer zu  $m(q, t)$  proportional retardierten Reibung.

Bis hierhin ist die Bewegungsgleichung für  $F(q, t)$  exakt, da der Memorykern noch nicht näher spezifiziert wurde. Bengtzeliu et al. [49] stellten in ihrer Arbeit die sogenannten (idealisierten) Modenkopplungsgleichungen vor, ein geschlossener Satz gekoppelter Gleichungen für  $F(q, t)$ , deren Lösung die volle Zeitabhängigkeit der intermediären Streufunktion beschreibt. Dabei wird  $m(q, t)$  über das Quadrat von  $F(q, t)$  ausgedrückt, d.h.

$$m(q, t) = \sum_{\vec{k}+\vec{p}=\vec{q}} V(\vec{q}; \vec{k}; \vec{p}) F(k, t) F(p, t), \quad (1.5)$$

wo die Punkte  $V(\vec{q}; \vec{k}; \vec{p})$  aus  $S(q)$  (und statischen Dreipunktskorrelationsfunktion) berechnet werden können. Man glaubt, daß diese selbstkonsistente Beschreibung die Dynamik für „kurze“ Zeitskalen, in der das Teilchen im „Käfig“ seiner Nachbarn gefangen ist, als auch den Prozess auf langen Zeitskalen, wenn das Teilchen aus diesem „Käfig“ ausbricht und diffusiv wird, korrekt wiedergibt.

Diese sogenannte Modenkopplungsnäherung (Gleichung (1.5)) beschreibt zusammen mit der Gleichung (1.2) einen nichtlinearen Rückkopplungsmechanismus in der Dichtefluktuation [52]. Das mikroskopische Bild dahinter ist folgendes: Jedes Teilchen befindet sich in einem „Käfig“, der aus seinen benachbarten Teilchen gebildet wird. Je tiefer die Temperatur, desto länger verbleibt das Teilchen in diesem Käfig. Aus diesem Käfig entweichen kann das Teilchen erst, wenn sich auch im benachbarten Käfig ein Teilchen befreien konnte, u.s.w. Das führt zu einem kollektiven Fließmechanismus. In der Näherung in Gleichung (1.4) findet man nun die Existenz eines Glasübergangs, bei dem die Teilchen sich bei einer kritischen Temperatur  $T_c$  derart behindern, daß sie nicht mehr aus ihren Käfigen entweichen können.

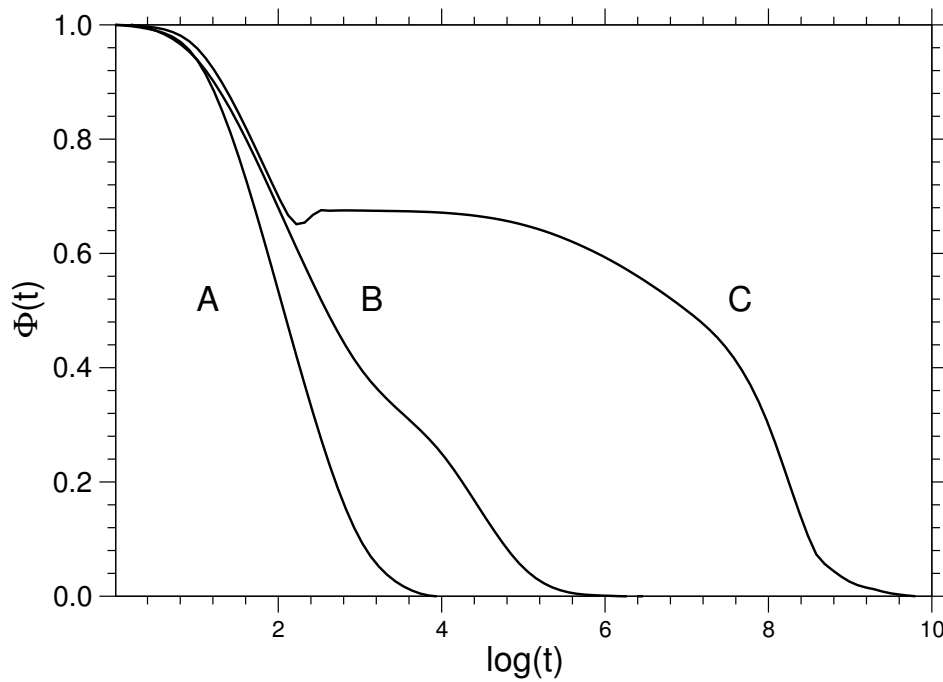
Aufgrund der Komplexität der Modenkopplungsgleichungen können Lösungen nur numerisch erlangt werden. Durch die Näherung, daß der Strukturfaktor durch eine  $\delta$ -Funktion bei dem Wellenvektor  $q_0$  gegeben ist<sup>6</sup>, kann man Gleichung (1.2) in eine einzige Korrelationsfunktion für  $q_0$  umwandeln. Über die Ersetzung  $\Phi(t) = F(q_0, t)/S(q_0)$  erhält man dann

$$\ddot{\Phi}(t) + \Omega^2\Phi(t) + \nu\dot{\Phi}(t) + \Omega^2 \int_0^t m[\Phi(t-t')] \dot{\Phi}(t') dt' = 0. \quad (1.6)$$

$m[\Phi]$  ist dabei ein Polynom niedriger Ordnung in  $\Phi$ . So eine Gleichung für wenige oder eine Korrelationsfunktion(en) nennt man *schematisches Modell*. Da über den Strukturfaktor  $S(q)$

---

<sup>6</sup>Wobei  $q_0$  die Position des Hauptpeaks von  $S(q)$  angibt.



**Abbildung 1.3.:** Schema der Zeitabhängigkeit der Korrelationsfunktion. Die verschiedenen Kurven entsprechen verschiedenen Temperaturen (A hohe Temperatur, C niedrige Temperatur).

eine Temperaturabhängigkeit auftritt, sind auch die Koeffizienten des Polynoms  $m[\Phi]$  temperaturabhängig.

Da in diesem schematischen Modell die Details der mikroskopischen Informationen beseitigt wurden, kann man damit nicht mehr experimentelle Daten quantitativ vergleichen. Durch die stark vereinfachte Komplexität kann man jedoch die analytische Untersuchung vereinfachen, um qualitative Vorhersagen über das Verhalten der intermediären Streufunktion zu machen. Bevor jetzt einige Vorhersagen der MCT besprochen werden, noch einige Anmerkungen zur idealisierten MCT:

- Wenn die Nichtlinearität, die im Memorykern  $m[\Phi]$  enthalten ist, eine gewisse Schwelle überschreitet, zerfällt die Lösung der (idealisierten) MCT-Gleichung für unendliche Zeiten nicht mehr auf 0, d.h. das System ist nicht mehr ergodisch. Dieser dynamische Übergang beschreibt einen idealen Glasübergang.
- Die Grenzen der idealisierten MCT finden sich in der Art der Transportprozesse. Oftmals treten (aktivierte) Prozesse auf, bei denen sich die Teilchen sprunghaft fortbewegen. Diese Prozesse werden *Hopping Prozesse* genannt, werden aber nicht durch die idealisierte MCT beschrieben. Es gibt Ansätze, die idealisierte Version der MCT um solche *Hopping Prozesse* zu erweitern<sup>7</sup>.

Im folgenden werden einige der Vorhersagen der MCT besprochen. Für einen besseren Überblick werden dem Leser z.B. die Publikationen [53, 54, 55, 56, 57, 58] empfohlen.

<sup>7</sup>Für weitere Betrachtungen um *hopping* erweiterter MCT sei der Leser auf [50] verwiesen.

Im weiteren richtet sich der Fokus auf die Vorhersagen der MCT bezüglich des Zerfalls der Korrelationsfunktionen (beispielhaft in Abbildung 1.3). Für kurze Zeiten erkennt man einen quadratischen Zerfall. Grund dafür ist die anfangs ballistische Bewegung der Teilchen. Für hohe Temperaturen (Kurve A) sieht man, daß dieses Verhalten schnell in ein Regime übergeht, daß man gut mittels eines exponentiellen Zerfall auf 0 beschreiben kann. Wird die Temperatur gesenkt (Kurven B und C), erkennt man zwischen ballistischen und exponentiellen Zerfall ein weiteres, auf intermediären Zeitskalen wirkendes Zerfallsregime, die  $\beta$ -Relaxation.<sup>8</sup> Für noch tiefere Temperaturen würde sich das Plateau auf immer längeren Zeitskalen ausbilden. Erst nach längeren Zeiten erreicht die Korrelationsfunktion das  $\alpha$ -Relaxations-Regime, in welchem sie vom Plateau auf 0 zerfällt.

Für die beiden Zeitregimes macht die MCT einige Vorhersagen, im speziellen auch zur funktionalen Form:

**$\alpha$ -Relaxation:** Eine der Hauptvorhersagen der MCT ist die Existenz einer kritischen Temperatur  $T_c$ , bei der die Selbstdiffusionskonstante  $D$  oder das Inverse der  $\alpha$ -Relaxationszeit  $\tau$  der Korrelationsfunktionen mit einem Potenzgesetz der Form

$$D \propto \tau^{-1} \propto (T - T_c)^\gamma \quad (1.7)$$

verschwindet, wobei der Exponent  $\gamma > 1.5$  ist. Eine andere Hauptvorhersage der MCT ist das sogenannte *Zeit-Temperatur-Superpositions-Prinzip* (ZTSP). Dieses sagt aus, daß die Korrelatoren für verschiedene Temperaturen mittels der Skalierung  $t/\tau$  auf eine Masterkurve  $\Psi$  abgebildet werden können. Dabei ist  $\tau$  die  $\alpha$ -Relaxationszeit, d.h.

$$\Phi(t) = \Psi(t/\tau(T)). \quad (1.8)$$

Desweiteren sagt die MCT voraus, daß diese Masterkurve gut mittels der *Kohlrausch-Williams-Watts* (KWW) Funktion

$$\Phi(t) = A \exp(-(t/\tau(T))^\beta) \quad (1.9)$$

gefittet werden kann. Die Korrelationsfunktionen teilen alle die Eigenschaft, daß die Korrelationszeiten bei  $T_c$  gemäß einem Potenzgesetz mit dem Exponenten  $\gamma$  divergieren. Dieser Exponent ist auch mit der  $\beta$ -Relaxation verbunden [50], d.h. man kann über die  $\alpha$ -Relaxation etwas über die  $\beta$ -Relaxation lernen und vice versa. Diese Verbindung ist über die Relation

$$\gamma = 1/(2a) + 1/(2b) \quad (1.10)$$

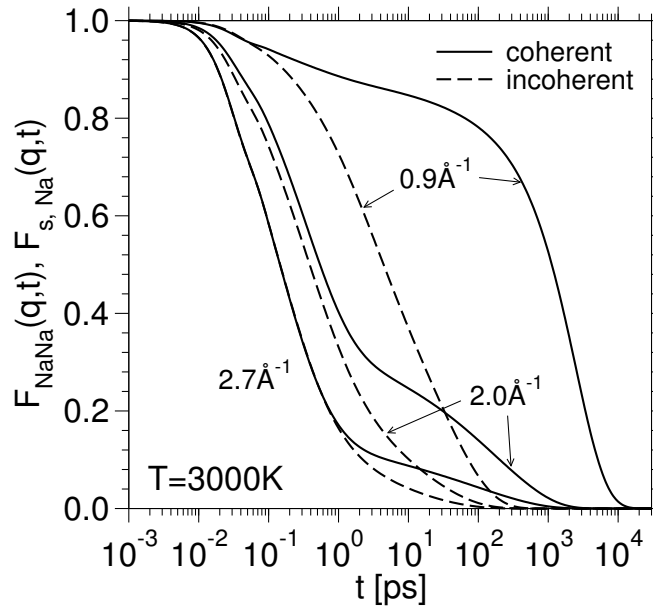
gegeben und verbindet den Exponenten, der die Zeitskala der  $\alpha$ -Relaxation bestimmt mit den für die  $\beta$ -Relaxation wichtigen Exponenten  $a$  und  $b$ .

**$\beta$ -Relaxation:** Dieses Regime ist als jenes charakterisiert, in dem der Korrelator nahe dem Plateau ist. Das späte  $\beta$ -Regime kann als Kurzzeitregime der  $\alpha$ -Relaxation verstanden werden. In der MCT ist die Kurzzeitentwicklung der Mastergleichung der  $\alpha$ -Relaxation bekannt und ergibt die folgende Formel für die intermediäre Streufunktion [60]

$$\Phi(q, t) = f_c(q) - h_c(q)(t/\tau)^b + h_2(q)(t/\tau)^{2b}. \quad (1.11)$$

---

<sup>8</sup>Eine anschauliche Beschreibung der mikroskopischen Vorgänge in den verschiedenen Regimes findet sich in Kapitel 2.2.2.1.



**Abbildung 1.4.:** Kohärente intermediäre Streufunktionen für die Na–Na–Korrelation und inkohärente Streufunktionen für Na bei den angegebenen  $q$ -Werten, berechnet durch Lösung der MCT–Gleichungen mit den partiellen statischen Struktur Faktoren für NS2 aus der Simulation bei  $T = 3000 \text{ K}$  als Input. (Zur Verfügung gestellt von Horbach [59].)

$f_c(q)$  ist die Höhe des Plateaus, auch nicht–Ergodizitätsparameter genannt.  $h_c(q)$  ist die kritische Amplitude die, genauso wie  $h_2(q)$ , nur von der Wellenzahl  $q$  abhängt (als auch von der Teilchensorte). Das erste Potenzgesetz in Gleichung (1.11) wird oft von *Schweidler Gesetz* und der Exponent  $b$  von *Schweidler–Exponent* genannt. Dieser ist nach MCT unabhängig vom Korrelator (d.h.  $q$ ). Für das Verhalten der Zeitskala der  $\beta$ –Relaxation  $t_\sigma$  (mit  $\sigma = (T - T_c)/T_c$ ) sagt die MCT voraus, daß  $t_\sigma$  sowohl oberhalb als auch unterhalb  $T_c$  für  $T \rightarrow T_c$  wie

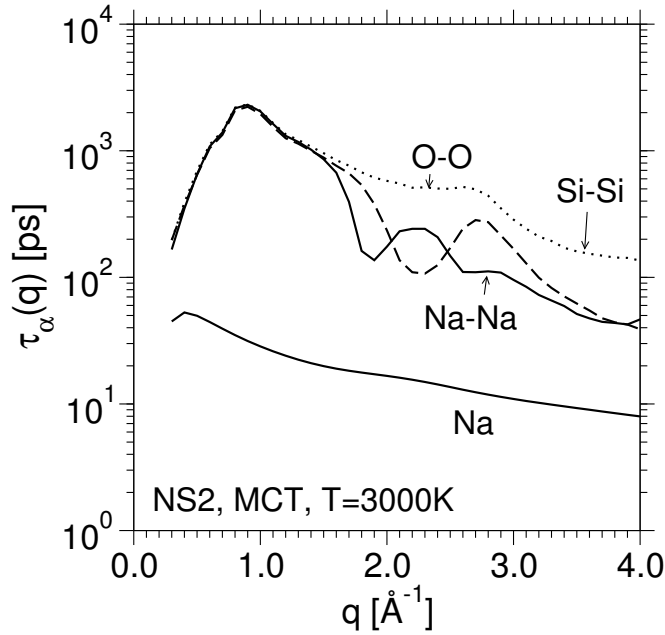
$$t_\sigma \propto |T - T_c|^{1/2a}, \quad (1.12)$$

divergiert, mit  $0 < a < 1/2$ . Lichtstreuungsexperimente haben die Validität der Divergenz der Zeitskala, wie in Gleichung (1.12) dargestellt, bestätigt [61, 62].

Die qualitativen Vorhersagen über das Skalenverhalten und die Form des Zerfalls der Korrelationen haben sich für viele unterkühlte Flüssigkeiten, auch z.B. für  $\text{SiO}_2$ , bestätigt (vgl. z.B. [6]). Die Frage, die sich in Bezug auf diese Arbeit stellt, ist, ob die Vorhersagen der MCT auch in binären Alkalisilikaten<sup>9</sup> funktionieren. Denn in diesen findet man eine hohe Mobilität der Alkaliionen<sup>10</sup>, die die Dynamik des Systems in zwei Zeitskalen unterteilt: Einmal in die „langsame“ Dynamik der sogenannten Matrixkomponenten Si und O, und zum anderen in die Dynamik der „schnellen“ Alkaliionen.

<sup>9</sup>Binär heißt hier, daß zwei Oxide im System vorhanden sind:  $\text{SiO}_2$  und  $\text{X}_2\text{O}$ , wobei X für eine Alkaliatomsorte steht.

<sup>10</sup>vgl. z.B. [6, 16, 51].



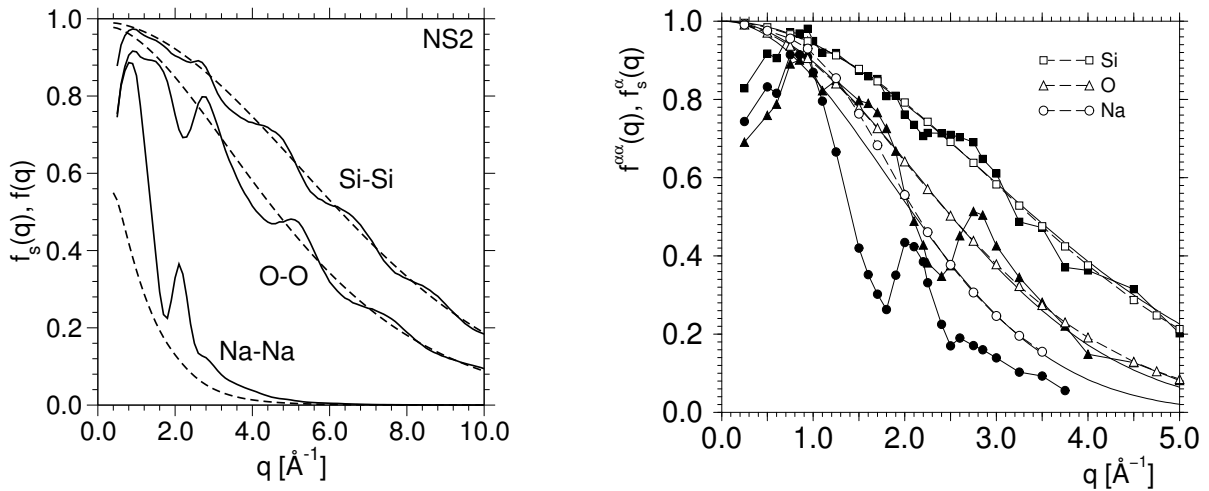
**Abbildung 1.5.:**  $q$ -Abhängigkeit der  $\alpha$ -Relaxationszeiten aus der MCT-Rechnung für NS2, extrahiert aus den kohärenten Streufunktionen für die Si-Si-, Na-Na-, und O-O-Korrelationen und aus den inkohärenten Streufunktionen für Na. Die Temperatur ist  $T = 3000$  K. (Zur Verfügung gestellt von Horbach [59].)

Ein Test, ob die MCT in oben beschriebener Form in der Lage ist, diese Zeitskalentrennung zu reproduzieren, wurde von Voigtmann [63] mit partiellen statischen Strukturfaktoren aus der Simulation zu Natriumdisilikat (NS2) von Horbach [59] vorgenommen. Resultate der Rechnungen für die (in)kohärenten Streufunktionen<sup>11</sup> ( $F_{s,Na}(q, t)$ )  $F_{NaNa}(q, t)$  sind in Abbildung 1.4 gezeigt. Die Rechnung liefert ein qualitativ ähnliches Verhalten wie die Simulation [16], speziell zerfällt  $F_{s,Na}(q, t)$  in der Nähe von  $q_1 = 0.95 \text{ \AA}^{-1}$  auf einer etwa zwei Größenordnungen kürzeren Zeitskala als  $F_{NaNa}(q, t)$ . Auch die  $q$ -Abhängigkeit der  $\alpha$ -Relaxationszeiten  $\tau_\alpha$  ist in qualitativer Übereinstimmung mit der Simulation (vgl. Abbildung 1.5). So sind die  $\tau_\alpha$  um  $q_1$  herum wie in der Simulation für alle kohärenten Korrelatoren identisch, wobei allerdings beachtet werden muß, daß es sich hierbei um MCT-Rechnungen bei 3000 K handelt, die mit Simulationsergebnissen bei 2100 K verglichen werden. Dieses Problem ist auch bei MCT-Rechnungen zu anderen Glasbildnern bekannt, z.B. bei binären Lennard-Jones-Mischungen, wo die kritische Temperatur  $T_c$  bei den MCT-Rechnungen [64] um einen Faktor 2 größer ist, als die aus der Simulation bestimmte.<sup>12</sup>

Auch wurde die Simulation von NS2 erfolgreich auf die Konsistenz mit den Vorhersagen der MCT zum Skalenverhalten überprüft. Für die Simulation ergaben sich die kritischen Exponenten  $\gamma = 2.87$  und  $b = 0.47$  und die kritische Temperatur  $T_c \approx 2000$  K. Die MCT-Rechnungen von Voigtmann lieferten dieselben Werte für die kritischen Exponenten, allerdings liegt das berechnete  $T_c$  bei 2670 K, also etwa ein Drittel höher als das aus der Simulation ermittelte  $T_c$ .

<sup>11</sup>Die Streufunktionen werden in Kapitel 2.2.2.3 definiert.

<sup>12</sup>Wobei in den MCT-Rechnungen die partiellen statischen Strukturfaktoren der Simulation als Grundlage verwendet wurden.



**Abbildung 1.6.:** Links: Debye-Waller-Faktoren (durchgezogene Linie) und Lamb-Möbbauser-Faktoren (gestrichelte Linie) aus der MCT-Rechnung für NS2. Rechts:  $q$ -Abhängigkeit der Debye-Waller-Faktoren  $f^{\delta\delta}(q)$  und Lamb-Möbbauser-Faktoren  $f_s^\delta(q)$  und (entsprechen Kurven mit geschlossenen und offenen Symbolen) für  $\delta \in \{\text{Si}, \text{O}, \text{Na}\}$ . Die breiten Kurven sind Gaußsche Fits zu den Lamb-Möbbauser-Faktoren der inkohärenten Funktionen. (Zur Verfügung gestellt von Horbach [59].)

Aber auch beim Vergleich der Debye-Waller- und Lamb-Möbbauser-Faktoren aus Simulation und MCT-Rechnung (siehe Abbildung 1.6) findet man ein qualitativ ähnliches Verhalten. Beispielsweise zerfällt der Debye-Waller- und der Lamb-Möbbauser-Faktor für Na-Na bzw. Na in beiden Fällen wesentlich schneller als für Si-Si und O-O bzw. Si und O, wobei dieser Effekt in der MCT-Rechnung jedoch ausgeprägter ist, als in der Simulation. Dort zerfallen die Lamb-Möbbauser-Faktoren für Si und O deutlich langsamer als in der Simulation, was bedeutet, daß Si und O in der MCT-Rechnung stärker lokalisiert sind. Ein möglicher Grund dafür könnte sein, daß in den MCT-Rechnungen statische Dreipunktskorrelationsfunktionen in den Vertices vernachlässigt wurden. Es wurde aber bereits von Sciortino and Kob [65] für  $\text{SiO}_2$  gezeigt, daß eine Berücksichtigung der Dreipunktskorrelationsfunktionen eine bessere Übereinstimmung mit der Simulation liefert.

Der vorgenommene Vergleich von Simulation und MCT-Rechnung für NS2 liefert Anhaltspunkte dafür, daß die MCT wesentliche Eigenschaften einer ionenleitenden Alkalischmelze qualitativ reproduziert. Er weist aber auch darauf hin, daß es mit der MCT in der aktuellen Form wichtige Eigenschaften gibt, die nicht richtig wiedergegeben werden. Beispielsweise folgt in der Simulation die Na-Diffusionskonstante  $D_{\text{Na}}$  im gesamten betrachteten Temperaturbereich einem Arrheniusverhalten gemäß  $D_{\text{Na}} \propto \exp(-0.93 \text{ eV}/(k_B T))$ . Die MCT sagt aber für  $D_{\text{Na}}$  ein Potenzgesetz gemäß  $D_{\text{Na}} \propto (T - T_c)^{-\gamma}$  voraus. Auch in der  $q$ -Abhängigkeit der Kohlrauschexponenten  $\beta(q)$  in Simulation [16, 66] und MCT-Rechnung gibt es starke Diskrepanzen. Die Simulation zeigt, daß der Kohlrauschexponent  $\beta(q)$  für die inkohärente Na-Streufunktion für  $q > 1.6 \text{ Å}^{-1}$  eine Konstante mit  $\beta = b = 0.47$  ist. Die MCT-Rechnung findet  $\beta = b$  erst für  $q > 9 \text{ Å}^{-1}$ . Diese Resultate zeigen, daß noch weitere Untersuchungen nötig sind, um die Gründe für die Diskrepanzen zwischen Simulation und MCT-Rechnungen aufzudecken, auf der anderen Seite haben die Untersuchungen von Horbach und Voigtmann gezeigt, daß einige qualitative Vorhersagen der MCT auch in ionenleitenden Silikaten gültig sind.



## 2. Simulationsdetails und Messgrößen

Dieses Kapitel soll einen Überblick über die technischen Details der Simulation und der verwendeten Messgrößen bieten. Es gliedert sich in eben jene beiden Teile: Zunächst eine Einführung in Molekular–Dynamik–Simulationen, des dabei verwendeten Potentials, die Umsetzung auf Parallelrechnern und zuletzt einen Überblick über die in der Arbeit simulierten Systeme.

Im zweiten Teil dieses Kapitels werden dann die in dieser Arbeit am häufigsten verwendeten Analysemethoden eingeführt. Diese gliedern sich in statische und dynamische Größen. In späteren Kapiteln werden, wenn nötig, noch zusätzlich einzelne spezielle Analysemethoden beschrieben.

### 2.1. Simulation: Durchführung und Methoden

#### 2.1.1. Molekular–Dynamik–Simulationen (MD)

Größe	Simulationseinheit
Längen	$\text{Å} = 10^{-10} \text{ m}$
Energien	$\text{eV} = 1.60217733 \cdot 10^{-19} \text{ J}$
Massen	$u = 1.6605402 \cdot 10^{-27} \text{ kg}$
Temperatur	$\frac{\text{eV}}{k_B} = \frac{\text{eV}}{1.380658 \cdot 10^{-23} \frac{\text{J}}{\text{K}}} = 1.160445 \cdot 10^4 \text{ K}$
Zeit	$\text{Å} \sqrt{\frac{u}{\text{eV}}} = 1.0180507 \cdot 10^{-14} \text{ s}$
Druck	$\text{eV}/\text{Å}^3 = 160.217733 \text{ GPa}$

**Tabelle 2.1.:** Übersicht der verwendeten Simulationseinheiten. Zahlenwerte aus [67] ( $k_B = 1.380658 \cdot 10^{23} \text{ J/K}$ ).

Molekulardynamik–Simulationen werden häufig dann verwendet, wenn, wie der Name es andeutet, die zeitliche dynamische Entwicklung des untersuchten Systems von Interesse ist. Denn die MD–Simulation liefert nicht nur statische Größen, sondern löst die Newtonschen Bewegungsgleichungen

$$\vec{F}_i = \dot{\vec{p}}_i \equiv m_i \cdot \ddot{\vec{r}}_i,$$



wobei  $i$  der Teilchenindex,  $\vec{p}_i$  der Impuls,  $\vec{F}_i$  die Kraft,  $\vec{r}_i$  der Ortsvektor und  $m_i$  die träge Masse<sup>1</sup> des Teilchens  $i$  ist. Dies liefert die klassischen Trajektorien der involvierten Teilchen und beschreibt so auch die Dynamik.

Diese Berechnung dieser Differentialgleichungen ist aber nur ein Problem, das sich bei der Simulation stellt. Ein anderes ist die Güte des verwendeten Potentials. Es geht darum, wie gut ein Potential Materialeigenschaften reproduziert. Dabei stoßen Paarpotentiale der Form, wie sie in dieser Arbeit verwendet wurden, in manchen Eigenschaften schnell an ihre Grenzen, in anderen wiederum sind die Resultate sehr gut (vgl. z.B. [6]). Elektronische Freiheitsgrade und Richtungsabhängigkeiten zum Beispiel werden darin nicht explizit berücksichtigt. Diese können zwar zum Potential dazumodelliert werden, machen dieses aber beliebig kompliziert und rechenaufwendig. Außerdem hat man nicht immer die Garantie, daß Potentiale, die komplexer sind, auch bessere Resultate liefern: Beispiele dafür finden sich in [69]. In dieser Arbeit wird das BKS–Paarpotential<sup>2</sup> mit Potentialen verglichen, die effektiv Mehrkörperwechselwirkungen berücksichtigen. Das wird durch konfigurationsabhängige Variation der Teilchenladung bzw. über die Berücksichtigung von Dipolmomenten versucht. Wie sich zeigte, lieferten diese Potentiale nicht immer bessere Ergebnisse, als das einfachere BKS–Paarpotential.

Es gibt zwar sowohl für die rein statische Berechnung (z.B. PIMC), als auch für die Dynamik (PIMD, Car–Parinello CPMD, etc.) Methoden der quantenmechanischen– bzw. Ab–Initio–Methoden Simulation, die allerdings wieder andere Probleme aufwerfen, wovon nicht das geringste der um mehrere Größenordnungen höhere Rechenaufwand ist. Darum werden auch PIMD und CPMD selten zur Untersuchung der Dynamik verwendet, da die Zeitskalen der Simulation, die in einer sinnvollen Rechenzeit abgedeckt werden können, schlicht zu kurz sind.<sup>3</sup>

Eine Möglichkeit, die Geschwindigkeit von klassischen Simulationen mit den Ergebnissen aus Ab–Initio Simulationen zu verbinden, ist das Anpassen von klassischen Potentialen an Ergebnisse von Ab–Initio–Simulationen. Ein Beispiel dafür ist das Paarpotential, das für die Molekulardynamik–Simulation (Kapitel 2.1.1.2) in dieser Arbeit verwendet wurde. Es basiert auf Arbeiten von Tsuneyuki et al. [70] und Habasaki and Okada [71]. Damit wurden Alkalisilikate der allgemeinen Form  $x(\text{Li}_2\text{O})(1-x)(\text{K}_2\text{O})(2\cdot\text{SiO}_2)$  simuliert.

### 2.1.1.1. Der Velocity–Verlet Algorithmus

In Kapitel 2.1.1.2 wird noch eine Verallgemeinerung des *Velocity–Verlet*–Algorithmus für das *NPH*–Ensemble behandelt. Hier soll zuerst der *Velocity–Verlet*–Algorithmus für das mikrokanonische Ensemble vorgestellt werden. Wie bereits in Kapitel 2.1.1 angesprochen, ist der Kern einer MD–Simulation das Lösen der Newtonschen Bewegungsgleichungen. Mittels des Zeitentwicklungsoperators  $e^{-i\mathcal{L}t}$  propagiert eine Phasenraumdichteverteilung  $\rho$  wie folgt (mit

---

<sup>1</sup>Die Identität von schwerer und träger Masse ist inzwischen bis auf eine Genauigkeit von  $\pm 5 \cdot 10^{-10}$  nachgewiesen [68].

<sup>2</sup>vgl. Infokasten auf Seite 34.

<sup>3</sup>Während in dieser Arbeit Systeme mit  $> 8000$  Teilchen bis zur Größenordnung  $80 \text{ ns}$  gerechnet wurden, werden bei der CPMD typischerweise  $\lesssim 100$  Teilchen bis zu  $10 \text{ ps}$  simuliert. Damit erreicht man aber in den hier untersuchten Systemen nicht das diffusive Regime.

Hamiltonian  $\mathcal{H}$ , Impulsen  $p_i$  und Orten  $q_i$ )<sup>4</sup>:

$$\rho(q, p, t) = e^{-i\mathcal{L}t} \rho(q, p, 0) \quad \text{mit} \quad -i\mathcal{L} = \{\mathcal{H}, \cdot\} = \sum_i \left( \underbrace{F_i}_{\frac{\partial \mathcal{H}}{\partial q_i}} \frac{\partial}{\partial p_i} - \underbrace{\frac{p_i}{m_i}}_{\frac{\partial \mathcal{H}}{\partial p_i}} \frac{\partial}{\partial q_i} \right). \quad (2.1)$$

Möchte man nun eine Aufspaltung des Liouville–Operators  $\mathcal{L}$  nach Ortsteil  $\mathcal{L}_q$  und Impulsteil  $\mathcal{L}_p$  vornehmen, also

$$-i\mathcal{L} = i\mathcal{L}_q + i\mathcal{L}_p, \quad (2.2)$$

so muß man dabei beachten, daß

$$\{\mathcal{L}_q, \mathcal{L}_p\} \neq 0 \quad (2.3)$$

gilt, d.h.  $\mathcal{L}_q$  und  $\mathcal{L}_p$  kommutieren nicht. Daher kann man keine exakte Aufspaltung der Exponentialfunktion in Gleichung (2.1) nach Impuls– und Ortsteil vornehmen, sondern erhält mittels

$$e^{ih(\mathcal{L}_q + \mathcal{L}_p)} = \underbrace{\prod_{i=1}^k e^{ic_i h \mathcal{L}_q} e^{id_i h \mathcal{L}_p}}_{S(h)} + \underbrace{\mathcal{O}(h^{k+1})}_{\text{Fehlerordnung}} \quad \text{mit} \quad \sum_i c_i = \sum_i d_i = 1 \quad (2.4)$$

eine Näherungslösung für  $e^{-i\mathcal{L}t}$  mit der Aufspaltung aus Gleichung (2.2).

Dabei ist folgendes zu beachten:

- $S(h)$  ist ein symplektischer Integrator, für den die Zeitumkehrinvarianz  $S(h)S(-h) \equiv \mathbb{1}$  gewährleistet sein muß.
- Daraus ergibt sich auch die Forderung, daß die Parameter  $c_i$  und  $d_i$  symmetrisch gewählt werden müssen (s.u.).
- Der Fehler  $\mathcal{O}$  der Ordnung  $k + 1$  in  $h$  entsteht dadurch, daß der Operator  $\mathcal{L}_q$  auf den Operator  $\mathcal{L}_p$  angewandt wird, obwohl diese nicht kommutieren (siehe Gleichung (2.3)).

Mittels dieser Entwicklung erhält man auch den *Velocity–Verlet–Algorithmus*. Dabei handelt es sich um ein Verfahren zweiter Ordnung ( $k = 2$ ), mit den symmetrisch gewählten Parametern

$$c_1 = 0, \quad d_1 = \frac{1}{2}, \quad c_2 = 1, \quad d_2 = \frac{1}{2}. \quad (2.5)$$

Damit erhält man den Integrator

$$S(h) = e^{\frac{1}{2}ih\mathcal{L}_p} e^{hi\mathcal{L}_q} e^{\frac{1}{2}ih\mathcal{L}_p}, \quad (2.6)$$

die sogenannte Trotter–Formel. Impuls– und Ortsteil wirken dabei wie folgt auf eine Phasenraumvariable  $A$  (mit  $q_0 = q(t = 0)$  und  $p_0 = p(t = 0)$ )

$$\begin{aligned} e^{ih\mathcal{L}_q} A(q_0, p_0) &= A\left(q_0 + \frac{p_0}{m}t, p_0\right) \\ e^{ih\mathcal{L}_p} A(q_0, p_0) &= A\left(q_0, p_0 + \underbrace{\dot{p}_0}_{F(q_0)}t\right), \end{aligned}$$

<sup>4</sup>ab hier wird die Betrachtung zur Vereinfachung 1–dimensional fortgesetzt.

wobei  $F(q)$  die Kraft am Ort  $q$  ist. Wendet man  $S(h)$  auf  $q$  und  $p$  an, so erhält man den *Velocity–Verlet–Algorithmus*:

$$\begin{aligned}
 \begin{pmatrix} q_0 \\ p_0 \end{pmatrix} &\longrightarrow \begin{pmatrix} q_0 \\ p_0 + \frac{1}{2}F(q_0)h \end{pmatrix} \\
 &\longrightarrow \begin{pmatrix} q_0 + \left[ p_0 + \frac{1}{2}F(q_0)h \right] \frac{h}{m} \\ p_0 + \frac{1}{2}F(q_0)h \end{pmatrix} \\
 &\longrightarrow \begin{pmatrix} q_0 + \frac{p_0}{m}h + \frac{1}{2m}F(q_0)h^2 \\ p_0 + \frac{1}{2}F(q_0)h + \frac{1}{2}F\left(q_0 + \frac{p_0}{m}h + \frac{1}{2m}F(q_0)h^2\right)h \end{pmatrix} \\
 &\longrightarrow \begin{pmatrix} q_0 + \frac{p_0}{m}h + \frac{1}{2m}F(q_0)h^2 \\ p_0 + \frac{1}{2}h [F(q_0) + F(q(h))] \end{pmatrix} \tag{2.7}
 \end{aligned}$$

Der Algorithmus sieht dann in der Form, in der er im Programm verwendet wird, folgend aus:

$$\begin{pmatrix} q_i(t+h) \\ v_i(t+h) \end{pmatrix} = \begin{pmatrix} q_i(t) + v_i(t)h + \frac{1}{2m}F_i(t)h^2 \\ v_i(t) + \frac{1}{2m}h [F_i(t) + F_i(t+h)] \end{pmatrix}, \tag{2.8}$$

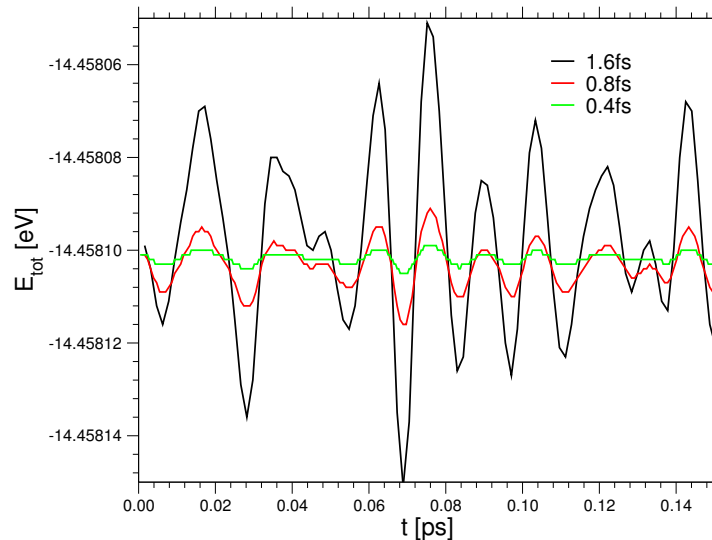
wobei  $i$  der Index für die Raumrichtungen ist. An diesem Punkt soll betont werden, daß der rechenaufwendigste Teil des Algorithmus die Berechnung der Kräfte ist.

Dieser Algorithmus zeigt einige für Simulationen sehr wichtige Eigenschaften:

- Der Algorithmus ist in dieser Form recht schnell. Ein Erhöhen der Näherungsordnung  $k = 2$  würde nur überproportional mehr Rechenzeit verbrauchen, da symplektische Integratoren höherer Ordnung die mehrfache Berechnung der Kräfte erfordern würden. Für die Zwecke der Simulation ist die Genauigkeit hoch genug.
- Der Algorithmus erhält die Energie. Zudem erhält der Algorithmus das Phasenraumvolumen, erfüllt also den Satz von Liouville und somit die Forderung aus der Hamiltonschen Mechanik.

Per Konstruktion ist der Algorithmus von quadratischer Ordnung in  $h$ . Variiert man  $h$ , so wächst die Fluktuationsamplitude der Energie  $\delta E = E - \langle E(t) \rangle$  proportional mit  $h^2$ . Diese Eigenschaft wird oft zum Testen von Potentialen auf die korrekte programmtechnische Umsetzungen verwendet. In dieser Arbeit wird dieser Test *Energietest* genannt (vgl. Abbildung 2.1). Er basiert darauf, daß für verschiedene Schrittweiten  $h$  die Fluktuationsamplitude zwar quadratisch wächst, der Mittelwert aber konstant bleiben muß. Daher wird anhand einer Rechnung mit verschiedenen Schrittweiten, wie in Abbildung 2.1 gezeigt, der Mittelwert der Energie und das Verhältnis von Amplituden in Abhängigkeit von der Schrittweite, geprüft.

In der Form von Gleichung (2.8) ist der Velocity–Verlet–Algorithmus nur für das mikrokanonische Ensemble ( $NVE$ ) zu verwenden. Im folgenden wird der Algorithmus noch erweitert, um auch die anderen Ensemble simulieren zu können, die für diese Arbeit noch wichtig sind.



**Abbildung 2.1.:** Energiefluktuation in Abhängigkeit von der Schrittweite. Wie in Kapitel 2.1.1.1 dargestellt, wächst die Amplitude der Energiefluktuation quadratisch mit der Variation der Schrittweite.

### 2.1.1.2. MD Simulationen in verschiedenen Ensembles

Um Vergleichbarkeit mit Ergebnissen aus Experimenten zu erreichen, wurden die Systeme zunächst im isobar/isothermen Ensemble ( $NpT$ ), bei  $p = 0$ , equilibriert. Die anderen Ensembles, isobar/isoenthalpisch ( $NpH$ ), kanonisch und mikrokanonisch erhält man als Spezialfälle des im folgenden beschriebenen ( $NpT$ )–Algorithmus, der eine Erweiterung des Velocity–Verlet–Algorithmus darstellt.

Für die Simulation der binären und ternären Systeme wurde ein *FORTTRAN*–Programm weiterentwickelt, das bereits in vielerlei Arbeiten zu Silikaten und Natriumsilikaten verwendet wurde (z.B. in [72, 6, 17, 73]). In diesem Programm war der sogenannte *Andersen–Thermostat*, der für eine **konstante Temperatur** sorgt, bereits implementiert. Um die Temperatur einzustellen, werden die Geschwindigkeiten aller Teilchen periodisch neu gezogen. Die Periodendauer wurde zwischen ca. 160 fs und 240 fs gewählt<sup>5</sup>, um das System in nicht zu kurzen Periodendauern zu stören (wird im folgenden noch ausgeführt).

Der Thermostat, der ein System mit Teilchen der Masse  $m_i$  auf die Geschwindigkeiten  $v_i$  einstellt, funktioniert wie folgt:

1. Zunächst werden für alle Teilchen Geschwindigkeiten  $u_i$  aus der Verteilung  $P(u_i) = \exp(-u_i^2/2)$  gezogen.
2. Um keine Drift im Gesamtsystem zu erhalten, z.B. um später Selbstdiffusionskonstanten bestimmen zu können, müssen die Geschwindigkeiten so gewählt werden, daß der Gesamtimpuls  $\sum_i m_i v_i = 0$  ist. Dafür werden zunächst die Geschwindigkeiten mit der Masse normiert  $u'_i = u_i/\sqrt{m_i}$ , um sie dann

<sup>5</sup>Entspricht 100 bis 150 Simulationsschritten.

Silizium	Si	28.086 u
Lithium	Li	6.941 u
Kalium	K	39.098 u
Sauerstoff	O	15.999 u

**Tabelle 2.2.:** Atomare Massen (aus [74]).

3. über  $u'_i = u'_i - \frac{\sum m_i u'_i}{Nm_i}$  zum Gesamtimpuls 0 zu normieren.
4. Die Temperatur  $T$  stellt man über den Gleichverteilungssatz  $3Nk_B T = \sum m_i v_i^2$  ein. Daher müssen im letzten Schritt die Geschwindigkeiten  $u'_i$  skaliert werden:  $v_i = u'_i \sqrt{\frac{3Nk_B T}{\sum m_i u'^2_i}}$ .

Einige Anmerkungen noch zum Thermostaten: Natürlich ist es unphysikalisch, die Geschwindigkeiten einfach stochastisch zu ziehen. Damit greift man mit der Frequenz der Anwendung des Thermostaten, also alle 160 bis 240 fs, in die Dynamik des Systems ein. Dadurch kann man in einem isothermen Ensemble mit dem Andersen–Thermostaten nur schwer Aussagen über die Dynamik machen, da diese bei jeder Thermostaten–Ankopplung gestört wird. Auf kurzen Zeitskalen zeigt sich die Ankopplung jedoch als unproblematisch, jedenfalls was die Form der Geschwindigkeitsverteilungen angeht. Das System „vergißt“ innerhalb von wenigen Femtosekunden den Eingriff des Thermostaten, d.h. die Geschwindigkeitsverteilungen stellen sich schnell wieder auf die Kräfte in Abhängigkeit der aktuellen Teilchenkonfiguration ein. Festgestellt wurde dies bei der Lösung des folgenden Problems.

Bei der Betrachtung der Geschwindigkeitsverteilungen stellte sich heraus, daß zwar der Gesamtimpuls aller Teilchen im Simulationssystem 0 ist, aber die einzelnen Teilchensorten Drift aufwiesen. Grund dafür ist die stark variierende Masse der verschiedenen Teilchensorten in dieser Arbeit (siehe Tabelle 2.2). Zwischen dem leichtesten verwendeten Element Li und dem schwersten K verändert sich die Masse um mehr als ein Faktor 4. Es passiert folgendes: Die Geschwindigkeiten werden zwar aus einer Gaußverteilung gezogen, aber da wir nur eine endliche Menge Zahlen ziehen, ist die Verteilung nicht perfekt gaußförmig und nicht exakt um 0 zentriert. Dies stellt bei hohen Massen nur ein kleines Problem dar. Wird aber nun der oben beschriebene Thermostat angewandt, so verschiebt sich die Verteilung der Geschwindigkeiten des leichteren Elements stärker weg von der symmetrischen Verteilung um 0. Folge ist ein gegenläufiger Drift verschiedener Teilchensorten. Um dies zu verhindern, wurde im Thermostaten eine stärkere Bedingung als die Impulserhaltung  $p = 0$  des Gesamtsystems eingeführt. Hier wird jetzt die Impulserhaltung  $p_\alpha = 0$  für jede Teilchensorte  $\alpha \in \{\text{Si, O, Li, K}\}$  gefordert. Dies wäre zwar nicht unbedingt nötig gewesen, da die Konfiguration innerhalb von ein bis zwei Simulationsschritten diese „Asymmetrie“ der Geschwindigkeiten vergessen hatte, entspricht aber eher dem Gedanken, der hinter dem Gesamtimpuls 0 im System steckt.

Nachdem nun der Velocity–Verlet–Algorithmus mittels des Andersen–Thermostaten um isotherme Eigenschaften erweitert wurde, soll nun die Erweiterung um **konstanten Druck** nach Andersen [75, 76] eingeführt werden. Aufgrund seiner Wichtigkeit für die Simulation wird dieser Algorithmus hier explizit ausgeführt.

Die Grundidee ist, daß ein System mit  $N$  Teilchen  $m_i$  vorliegt, das mit einem beweglichen, isotropen Kolben der Masse  $M$  gekoppelt ist. Dieser umgibt das gesamte System und variiert

das Volumen, welches hier also keine Konstante ist, sondern sich mittels Bewegungsgleichungen entwickelt.

Da das Volumen  $V$  die konjugierte thermodynamische Größe zum Druck  $p$  ist, kann man diesen mittels  $V$  regulieren. Um den Druck auf einen bestimmten Wert einzustellen, nimmt man einen externen Druck  $p_E$  an, der von außen auf das Volumen  $V$  einwirkt. Der Lagrangian wird wie folgt angesetzt

$$\mathcal{L} = \sum_{i=1}^N \frac{1}{2} m_i \dot{\vec{r}}_i^2 + \frac{1}{2} M \dot{V}^2 - U(\{\vec{r}_i\}) - p_E V \quad (2.9)$$

mit  $m_i$ ,  $\vec{r}_i$  und  $\vec{v}_i$  die Massen, Orte und Geschwindigkeiten der Teilchen,  $U(\{\vec{r}_i\}) = \sum_{i<j} u(r_{ij})$  dem Gesamtpotential ihrer Wechselwirkung.  $M$ ,  $V$  und  $\dot{V}$  sind die Masse des Kolbens, das Volumen des Systems sowie dessen Ableitung nach der Zeit. Für einen konstanten Druck muß das Volumen variieren können. Das bedeutet, es müssen Bewegungsgleichungen für das Volumen aufgestellt werden. Dafür müssen die Teilchenkoordinaten  $\{\vec{r}_i\}$  mit dem Volumen  $V$  die in Gleichung (2.9) noch ungekoppelt sind, gekoppelt werden. Dafür führte Andersen skalierte Koordinaten  $\{\vec{\rho}_i\}$  ein:

$$\vec{r}_i = V^{1/3} \vec{\rho}_i.$$

Damit erhält man

$$\mathcal{L} = \sum_{i=1}^N \frac{1}{2} m_i \dot{\vec{\rho}}_i^2 V^{2/3} + \frac{1}{2} M \dot{V}^2 - U(\{\vec{\rho}_i V^{1/3}\}) - p_E V \quad (2.10)$$

und daraus erhält man die Lagrange-Gleichungen

$$\begin{aligned} \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\rho}_k} &= m_k V^{2/3} \dot{\rho}_k + m_k \frac{2}{3} \dot{V}^{-1/3} \dot{V} \dot{\rho}_k \\ &= \frac{\partial \mathcal{L}}{\partial \rho_k} = F_k(\{\vec{r}_i\}) V^{1/3} \end{aligned} \quad (2.11)$$

$$\begin{aligned} \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{V}} &= M \dot{V} \\ &= \frac{\partial \mathcal{L}}{\partial V} = \frac{1}{3} V^{-1/3} \left( \sum_i m_i \dot{\rho}_i^2 + \sum_{i<j} \vec{r}_{ij} \cdot \vec{F}_{ij} V^{-2/3} \right) - p_E \\ &= \left[ \frac{1}{3L} \left( \sum_i m_i \dot{\rho}_i^2 \right) + \frac{1}{3V} \sum_{i<j} \vec{r}_{ij} \cdot \vec{F}_{ij} \right] - p_E \\ &= p(t) - p_E \end{aligned} \quad (2.12)$$

Dabei sind  $i, j \in \{1, \dots, N\}$  Teilchenindizes und  $k \in \{1, \dots, 3N\}$  ein gemeinsamer Index über Teilchen und Richtungen.  $F_k = F_i^\mu$  ist die Kraft auf das Teilchen  $i$  mit dem Richtungsindex  $\mu$ . Desweiteren ist  $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$  und  $\vec{F}_{ij} = \vec{F}_i - \vec{F}_j$ . Es sei betont, daß  $p(t)$  ein instantaner Druck ist. Der thermodynamische Druck ergibt sich hierbei durch thermische Mittelung, also  $\langle \dots \rangle$  gemäß dem Virialsatz

$$pV = Nk_B T + \frac{1}{3} \left\langle \sum_{i=1}^N \vec{r}_i \cdot \vec{F}_i \right\rangle \quad (2.13)$$

mit der Kraft  $\vec{F}_i = \sum_j \vec{F}_{ij}$  auf das Teilchen  $i$ . Die Bewegungsgleichungen sind also

$$\ddot{\rho}_i(t) = \frac{\vec{F}_i(t)}{m_i L(t)} - \frac{2}{3} \dot{\rho}_i(t) \frac{\dot{V}(t)}{V(t)} \quad (2.14)$$

$$\ddot{V}(t) = (p(t) - p_E) / M \quad (2.15)$$

mit  $L = V^{1/3}$ .

Nun müssen die Bewegungsgleichungen geschickt integriert werden. Dabei wird hier der Weg von Fox and Andersen [76] übernommen. Die Integration für die reskalierten Koordinaten und das Volumen erhält man mittels einer Taylor-Entwicklung bis zur zweiten Ordnung in  $h$ , wobei  $h$  das Zeitintervall zwischen den Zeitschritten  $n$  und  $n + 1$  ist. Also gilt mit Gleichung (2.14) und Gleichung (2.15)

$$\vec{\rho}_i^{n+1} = \vec{\rho}_i^n + h \dot{\vec{\rho}}_i^n + \frac{1}{2} \frac{h^2 \vec{F}_i^n}{m_i L^n} - \frac{h^2}{3} \dot{\vec{\rho}}_i^n \frac{\dot{V}^n}{V^n} \quad (2.16)$$

$$V^{n+1} = V^n + h \dot{V}^n + \frac{1}{2} h^2 (p_n - p_E) / M. \quad (2.17)$$

Um die Reversibilität des Velocity-Verlet-Algorithmus zu erhalten, d.h.  $(\vec{\rho}_i^{n+1}, V^{n+1}, \vec{\rho}_i^n, V^n, h) \mapsto (\vec{\rho}_i^n, V^n, \vec{\rho}_i^{n+1}, V^{n+1}, -h)$  zu gewährleisten, wird die erste Ableitung symmetrisch in  $n$  und  $n + 1$  integriert, d.h.

$$\dot{\rho}^{n+1} = \dot{\rho}^n + \frac{h}{2} (\ddot{\rho}^n + \ddot{\rho}^{n+1}) \quad (2.18)$$

$$\dot{V}^{n+1} = \dot{V}^n + \frac{h}{2} (\ddot{V}^n + \ddot{V}^{n+1}). \quad (2.19)$$

Hier stößt man aber auf ein Problem, denn  $\ddot{\rho}^{n+1}$  hängt nach Gleichung (2.14) von  $\dot{V}^{n+1}$  ab.  $\dot{V}^{n+1}$  wiederum hängt über  $p$  von der Menge  $\{\dot{\rho}^{n+1}\}$  ab (Gleichung (2.15)). Im Folgenden wird diese (und analoge) Menge(n) von  $\rho$  mit  $\dot{\rho}^{n+1}$  abgekürzt. Da die Gleichungen gekoppelt sind, wird die Lösung durch iterative Integration genähert. Daher wird statt einer exakten Integration von  $\dot{V}^{n+1}$  zunächst nur eine approximative Lösung mittels  $\dot{\rho}^n$  berechnet:

$$\dot{V}^{n+1} = \dot{V}^n + \frac{h}{2} \left\{ \ddot{V}(\rho^n, \dot{\rho}^n, V^n) + \ddot{V}(\rho^{n+1}, \dot{\rho}^{n+1}, V^{n+1}) \right\}, \quad (2.20)$$

$$\dot{V}_{\text{app}}^{n+1} = \dot{V}^n + \frac{h}{2} \left\{ \ddot{V}(\rho^n, \dot{\rho}^n, V^n) + \ddot{V}(\rho^{n+1}, \dot{\rho}^n, V^{n+1}) \right\}. \quad (2.21)$$

Also erhält man

$$\dot{V}_{\text{app}}^{n+1} = \dot{V}^n + \frac{h}{2} \left\{ \frac{p^n - p_E}{M} + \frac{p(\rho^{n+1}, \dot{\rho}^n, V^{n+1}) - p_E}{M} \right\}. \quad (2.22)$$

Mittels dieser Näherung berechnet man nun  $\dot{\rho}_i^{n+1}$

$$\begin{aligned} \dot{\rho}_i^{n+1} &= \dot{\rho}_i^n + \frac{h}{2} \left\{ \ddot{\rho}_i + \ddot{\rho}_i(\rho^{n+1}, \dot{\rho}^n, V^{n+1}, \dot{V}_{\text{app}}^{n+1}) \right\} \\ &= \dot{\rho}_i^n + \frac{h}{2} \left\{ \frac{\vec{F}_i^n}{m_i L^n} - \frac{2}{3} \dot{\rho}_i^n \frac{\dot{V}^n}{V^n} + \frac{\vec{F}_i^{n+1}}{m_i L^{n+1}} - \frac{2}{3} \dot{\rho}_i^{n+1} \frac{\dot{V}_{\text{app}}^{n+1}}{V^{n+1}} \right\}. \end{aligned} \quad (2.23)$$

Diese Gleichung löst man nun nach  $\dot{\rho}_i^{n+1}$  auf und entwickelt  $\left(1 + \frac{h}{3} \frac{\dot{V}_{\text{app}}^{n+1}}{V^{n+1}}\right)^{-1}$  bis zur zweiten Ordnung in  $h$ . Unter Berücksichtigung von Termen bis zur zweiten Ordnung von  $h$  erhält man dann

$$\dot{\rho}_i^{n+1} = \dot{\rho}_i^n + h \left\{ \frac{1}{2} \left( \frac{\vec{F}_i^n}{m_i L^n} + \frac{\vec{F}_i^{n+1}}{m_i L^{n+1}} \right) - \frac{1}{3} \left( \frac{\dot{V}^n}{V^n} + \frac{\dot{V}_{\text{app}}^{n+1}}{V^{n+1}} \right) \cdot \dot{\rho}_i^n \right\} 1 - \frac{h}{3} \frac{\dot{V}_{\text{app}}^{n+1}}{V^{n+1}}. \quad (2.24)$$

Die so erhaltenen reskalierten Geschwindigkeiten  $\{\dot{\rho}_i^{n+1}\}$  setzt man nun noch einmal in Gleichung (2.20) ein

$$\dot{V}^{n+1} = \dot{V}^n + \frac{h}{2} \left\{ \frac{p^n - p_E}{M} + \frac{p(\rho^{n+1}, \dot{\rho}^{n+1}, V^{n+1}) - p_E}{M} \right\}. \quad (2.25)$$

Damit ist der Simulationsalgorithmus definiert. Man startet ihn mit einer Anfangskonfiguration  $(\{\vec{\rho}_i\}, V)$  und Anfangsgeschwindigkeiten  $(\{\dot{\rho}_i\}, \dot{V})$  und integriert dann sukzessive von Zeitschritt  $n \rightarrow n + 1$  mit den Gleichungen (2.16), (2.17), (2.22), (2.24) und (2.25).

Der Algorithmus ist in dieser Form reversibel in der Zeit (vgl. dazu [72]).

### 2.1.1.3. Anfangsbedingungen

Nun stellt sich aber das Problem, wie man denn eine solche Anfangskonfiguration  $\{\vec{\rho}_i\}$  erhält. Im wesentlichen gibt es zwei in der Praxis häufig verwendete Möglichkeiten. Man kann einerseits die verwendeten Teilchen in der Kristallstruktur (wenn vorhanden) anordnen und diesen Kristall dann mittels des Thermostaten erhitzen, um dann die Schmelze zu erhalten, die man gegebenenfalls zu Glas abkühlen kann. Eine andere Möglichkeit ist die Wahl zufällig verteilter Teilchen. Natürlich entstehen bei solchen Zufallskonfigurationen sehr große repulsive Kräfte durch mögliche kurze Abstände zwischen den Atomen [77]. Dies kriegt man man mit sehr kurzen Zeitschritten Größenordnung  $\sim 0.1$  fs und häufigem Ankoppeln des Wärmebads ( $NpT$  bzw.  $NVT$ ) in der Simulation schnell innerhalb Zeiten  $\ll 1$  ps in den Griff, so daß das System aufgrund der riesigen Kräfte nicht „explodiert“, d.h. das die Teilchen nicht innerhalb eines Schrittes schon die Simulationsbox verlassen (was trotz periodischer Randbedingungen passieren kann, da aus Optimierungsgründen das Programm nur mit Schrittweiten bis zu einer Boxlänge klarkommt).

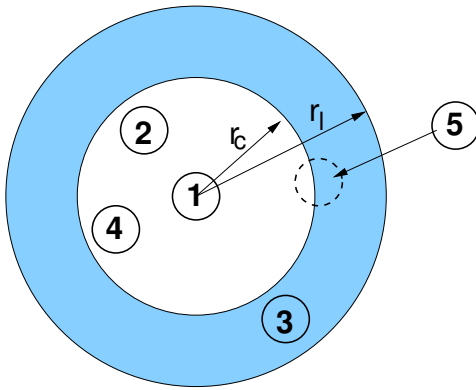
Danach kann man die Flüssigkeit auf eine hohe Temperatur, in dieser Arbeit meist 4000 K, äquilibrieren. Die dabei entstehenden Konfigurationen stellen dann auch die Ausgangskonfigurationen für Läufe bei niedrigeren Temperaturen dar. Die Schrittweite muß dabei zwei gegenläufigen Bedingungen genügen: Die Schrittweite sollte möglichst groß sein, um Rechenzeit zu sparen und die Schrittweite muß so klein sein, daß die Energie im Rahmen des Diskretisierungsfehlers in mikrokanonischen Läufen ( $NVE$ ) erhalten bleibt. Dabei stellt sich die Energieerhaltung als das strengere Kriterium heraus, die einerseits über den sog. *Energie-test*<sup>6</sup> überprüft wird, andererseits über die Konstanz der Energie über den Verlauf des gesamten mikrokanonischen Produktionslaufs.

<sup>6</sup>vgl. Kapitel 2.1.1.1.









**Abbildung 2.4.:** Cutoff Kugel mit Radius  $r_c$  sowie „Haut“ mit Dicke  $r_l - r_c$  (hellblau). Die Teilchen 2 und 4 wechselwirken mit Teilchen 1. Teilchen 3 ist zwar noch in der Nachbarliste von Teilchen 1, wechselwirkt aber nicht mit ihm. Wenn ein Teilchen (hier 5) sich um mehr als  $r_l - r_c$  bewegt hat, müssen die Nachbarlisten erneuert werden, da das Teilchen den Radius  $r_c$  durchdrungen haben könnte.

### 2.1.1.6. Nachbarlisten

Bei großen Systemen, ist es vom Standpunkt des Rechenaufwandes nicht effizient, alle Teilchenpaare, egal wie weit voneinander entfernt, zu überprüfen, ob sie in Wechselwirkung miteinander stehen. Die nötige Rechenzeit lässt sich durch die Verwendung von *Nachbarlisten* verringern.

In der sogenannten *Verletschen Nachbarliste* führt man eine Cutoff-Kugel mit dem Radius  $r_c$  (siehe Abbildung 2.4) um jedes Teilchen ein. Diese ist von einer „Haut“ der Dicke  $r_l - r_c$  umgeben. Sinn dieser Haut ist es, die Neuberechnung der Nachbarlisten in jedem einzelnen Simulationsschritt zu vermeiden. Erst wenn sich irgendein Teilchen der gesamten Konfiguration um mehr als die Dicke der „Haut“ bewegt hat, so werden alle Nachbarlisten neu berechnet, da dann ein Teilchen von außerhalb  $r_l$  es bis zur Wechselwirkungsreichweite  $r_c$  geschafft haben könnte.

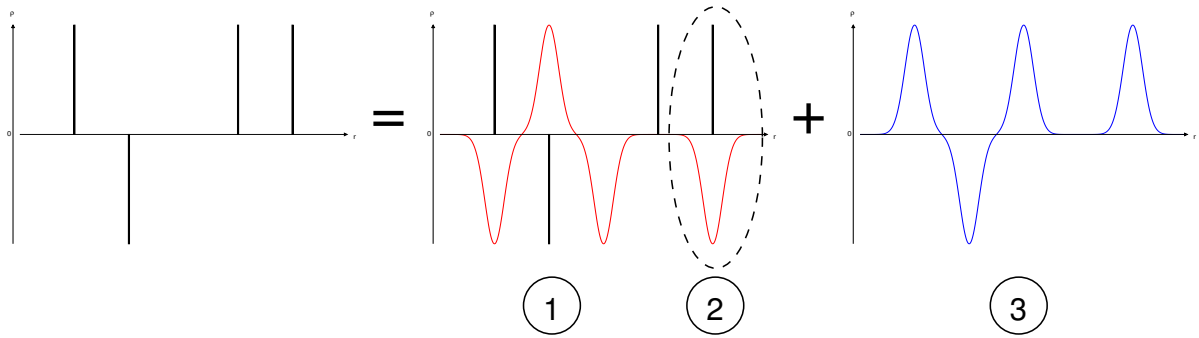
Alle Teilchen innerhalb des Radius  $r_l$  werden auf Wechselwirkung überprüft. Beiträge liefern aber nur die Teilchen innerhalb  $r_c$ .

Die Dicke der „Haut“ ist über  $r_l$  frei wählbar. Hier muß allerdings eine Balance zwischen folgenden Faktoren gefunden werden:

- Ist  $r_l \gg r_c$ , so werden Nachbarlisten-Updates seltener, aber es müssen viele Teilchen auf Wechselwirkung hin geprüft werden, die weiter als  $r_c$  entfernt sind.
- Ist  $r_l - r_c$  sehr klein, so werden nur wenige Teilchen unnötig auf Wechselwirkung hin geprüft, aber die Nachbarlisten müssen häufig erneuert werden.

### 2.1.1.7. Ewald-Summation

Der Coulombanteil des verwendeten Potentials (später in Kapitel 2.1.2.1) stellt eine Herausforderung an die praktische Umsetzung der Simulation dar. Da es sich nicht um einen rein kurzreichweitigen Anteil handelt (siehe auch Kapitel 2.1.1.4), sondern das Coulomb-Potential langreichweitig ist, wechselwirkt eine Konfiguration von Teilchen  $\{\vec{r}_i\}_{i=1,\dots,N}$  bei periodischen Randbedingungen (hier: kubische Simulationsbox, Kantenlänge  $N$ ) nicht nur mit allen



**Abbildung 2.5.:** Konzept der Ewald-Methode: Links die Punktladungen. Diese werden ersetzt durch Punktladungen mit entgegengesetzt Gaußförmig verteilten Ladungen (rot), die zur Folge haben, daß nur noch einfache kurzreichweitige Wechselwirkungen übrig bleiben, da die resultierenden Reihen schnell konvergieren. Als Korrektur muß man jedoch die Gaußfunktionen dazuaddieren (blau), und die Selbstwechselwirkung zwischen Ladung und Abschirmladung (gestricheltes Oval) abziehen.

Teilchen  $j \neq i$ , sondern auch mit allen periodischen Bildern der Teilchen  $\{\vec{r}_i + \vec{n}L | \vec{n} \in \mathbb{Z}^3, \vec{n} \neq \vec{0}\}_{i=1, \dots, N}$ . Der Coulombanteil nimmt daher folgende Form an:

$$V_C(\{\vec{r}_i\}) = \frac{1}{2} \sum_{\vec{n} \in \mathbb{Z}^3} \sum_{\substack{i,j=1 \\ i \neq j \text{ fr } \vec{n}=\vec{0}}}^N \frac{q_i q_j e^2}{|\vec{r}_{ij} + \vec{n}L|}. \quad (2.26)$$

Die numerische Auswertung von Gleichung (2.26) ist als *klassisches Madelung-Problem* [80] bekannt. Die Gittersumme des Coulombpotentials konvergiert nur langsam und ist nur bedingt konvergent, d.h. abhängig von der Summationsreihenfolge. Desweiteren ist ein sphärischer Cutoff, wie in dieser Arbeit für kurzreichweitige Anteile verwendet (vgl. Kapitel 2.1.1.6), ungeeignet, da das Sub-System innerhalb der Kugel praktisch nie ladungsneutral sein wird (vgl z.B. [81, 82]).

Dieses Problem löst ein Ansatz von Wolf et al. [83]. Ladungsneutralität wird dort unabhängig vom Cutoff-Radius des Coulomb-Potentials erreicht. Die Argumentation lautet so, daß jede Nettoladung in der lokalen, sphärischen Umgebung der Ionen aus dem Aufbrechen der Moleküle nahe der Oberfläche der Cutoff-Kugel resultiert. Die Nettoladung wird als auf der Oberfläche sitzend angenommen und kann als solche durch einen einfachen Ausdruck neutralisiert werden. Obwohl einige Tests mit dieser Methode durchgeführt wurden (z.B. [84]), gibt es keinen systematischen Zugang zu einer Abschätzung des Fehlers, den man bei Anwendung dieser Methode macht.

Daher wird dieses Problem in dieser Arbeit durch die Verwendung der *Methode der Ewaldsummen* gelöst, die 1921 von Ewald [85] veröffentlicht wurde. Sie gilt als Standardverfahren<sup>8</sup>, um Probleme bedingter Konvergenz im Falle periodischer Systeme/Randbedingungen zu lösen. Der Ansatz ist folgender: Die bedingt konvergente Summe über  $1/r$  wird in zwei schnell konvergierende Reihen zerlegt, zum einen eine Summe kurzreichweitiger Potentiale im Ortsraum, welche problemlos abgeschnitten werden, zum anderen eine Summe über reziproke Gittervektoren im Fourierraum.

<sup>8</sup>Denn im Gegensatz zur Wolf-Methode ist der Fehler hier bekannt.

Die zentrale Idee der Ewaldmethode besteht darin, Punktladungen als eine Menge abgeschirmter Ladungen abzüglich einer stetigen periodisch variierenden Hintergrundladungsverteilung anzusehen (anschaulich: Abbildung 2.5).

Die Ladungsdichte nach Gleichung (2.26) läßt sich mathematisch auch als eine Summe von  $\delta$ -Funktionen darstellen, bei der jeder Beitrag mit  $1/r$  abfällt. Zu jedem Punktladungsbeitrag wird dann eine entgegengesetzt geladene Gaußsche Ladungswolke addiert (siehe Abbildung 2.5), wodurch jeder Ladungsbeitrag eine in  $r$  schnell abfallende Funktion wird. Diese künstlich eingeführte Abschirmladung muß natürlich korrigiert werden. Für ein Punktteilchen mit der Ladung 1 sieht der Ansatz wie folgt aus:

$$\begin{aligned} \frac{1}{r} &= \frac{1}{r} - \frac{2}{\sqrt{\pi}} \int_0^\alpha d\rho \exp(-r^2 \rho^2) + \frac{2}{\sqrt{\pi}} \int_0^\alpha d\rho \exp(-r^2 \rho^2) \\ &= \frac{\operatorname{erfc}(\alpha r)}{r} + \frac{2}{\sqrt{\pi}} \int_0^\alpha d\rho \exp(-r^2 \rho^2). \end{aligned} \quad (2.27)$$

Der Parameter  $\alpha$  ist frei wählbar<sup>9</sup> und  $\operatorname{erfc}$  ist die komplementäre Fehlerfunktion,

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty d\rho e^{-\rho^2}.$$

Mit  $r := |\vec{r}_{ij} + \vec{n}L|$  folgt

$$V_C = S_1 + S_2 \quad \text{mit} \quad (2.28)$$

$$S_1 = \frac{1}{2} \sum_{\vec{n} \in \mathbb{Z}^3} \sum_{\substack{i, j = 1 \\ i \neq j, \vec{n} = \vec{0}}}^N \frac{q_i q_j e^2}{|\vec{r}_{ij} + \vec{n}L|} \operatorname{erfc}(\alpha |\vec{r}_{ij} + \vec{n}L|) \quad (2.29)$$

$$S_2 = \frac{1}{2} \sum_{\vec{n} \in \mathbb{Z}^3} \sum_{\substack{i, j = 1 \\ i \neq j, \vec{n} = \vec{0}}}^N \frac{2q_i q_j e^2}{\sqrt{\pi}} \int_0^\alpha d\rho \exp(-|\vec{r}_{ij} + \vec{n}L|^2 \rho^2), \quad (2.30)$$

wobei  $S_1$  bereits schnell konvergiert. Werden zu  $S_2$  noch die Terme für  $i = j$  bei  $\vec{n} = \vec{0}$  hinzugefügt, so erhält man für die zugehörige kompensierende Ladungsverteilung eine stetig variierende periodische Funktion, die sich über schnell konvergierende Fourierreihen berechnen läßt. Da die Gittersumme einer Funktion  $F$  gemäß der Poissonschen Summationsformel ( $\Theta$ -Transformation [86]) gleich der Gittersumme über deren Fouriertransformierte  $\hat{F}$  ist, also  $\sum_{\vec{n} \in \mathbb{Z}^3} F(\vec{n}) = \sum_{\vec{m} \in \mathbb{Z}^3} \hat{F}(\vec{m})$ , ergibt sich für  $S_2$ :

$$S_2 = \frac{1}{2} \sum_{i, j = 1}^N \frac{2q_i q_j e^2}{\sqrt{\pi}} \int_0^\alpha d\rho \sum_{\vec{m} \in \mathbb{Z}^3 \setminus \vec{m} = \vec{0}} \int d\vec{n} \exp(-|\vec{r}_{ij} + \vec{n}L|^2 \rho^2) \exp(-2\pi \vec{n} \cdot \vec{m}) - S_{2,S} \quad (2.31)$$

mit dem Selbstanteil  $S_{2,S}$ , der die Terme für  $i = j$  bei  $\vec{n} = \vec{0}$  enthält, die zur Anwendung der Poissonschen Summationsformel hinzuaddiert wurden. Die Lösung der Integrale ergibt dann

<sup>9</sup>Die Wahl des Parameters  $\alpha$  wird später in diesem Kapitel behandelt.

[87]:

$$S_2 = \frac{1}{2\pi L} \sum_{\vec{m} \in \mathbb{Z}^3 \setminus \vec{m} = \vec{0}} \frac{\exp\left(-\left(\frac{\pi m}{\alpha L}\right)^2\right)}{m^2} \sum_{i,j=1}^N q_i q_j e^2 \exp\left(i \frac{2\pi}{L} \vec{r}_{ij} \cdot \vec{m}\right) - S_{2,S}. \quad (2.32)$$

Daß hier die Terme für  $\vec{m} = \vec{0}$  ausgeschlossen werden, wird zwar oft stillschweigend angenommen, ist jedoch nicht trivial. Es liegt an der bedingten Konvergenz der Ewald–Summe: Es mag auf den ersten Blick seltsam erscheinen, aber das Potential eines unendlichen, periodischen Systems geladener Teilchen hängt von den Randbedingungen im Unendlichen ab, d.h. vom Medium, in dem das System eingebettet ist. Ist die Dielektrizitätskonstante des Mediums endlich, so treten an der Grenze des simulierten Systems, d.h. im Unendlichen, Oberflächenladungen auf. Dies wiederum führt zu einem depolarisierendem Feld und somit zu einem zusätzlichen Potentialterm.

Nimmt man jedoch, wie hier gemacht, eine unendlich große Dielektrizitätskonstante an, also einen Leiter, verschwindet das Feld und somit kann auch der Term für  $\vec{m} = \vec{0}$  in Gleichung (2.31) vernachlässigt werden (siehe auch [88]).

Den Selbstanteil  $S_{2,S}$  erhält man über den Limes  $\epsilon_i := r_{ij} \rightarrow 0$ :

$$S_{2,S} = \frac{1}{2} \sum_{i=1}^N \frac{2q_i^2 e^2}{\sqrt{\pi}} \lim_{\epsilon_i \rightarrow 0} \int_0^\alpha d\rho \exp(-\epsilon_i^2 \rho^2) \quad (2.33)$$

$$= \frac{1}{2} \sum_{i=1}^N q_i^2 e^2 \lim_{\epsilon_i \rightarrow 0} \frac{1 - \operatorname{erfc}(\alpha \epsilon_i)}{\epsilon_i} \quad (2.34)$$

$$\simeq \frac{\alpha}{\sqrt{\pi}} \sum_{i=1}^N q_i^2 e^2 \quad (2.35)$$

mit  $\operatorname{erfc}(x) = 1 - \frac{2}{\sqrt{\pi}}x + \mathcal{O}(x^3)$ .

Für die praktische numerische Umsetzung ist es nötig, die Reihen  $S_1$  und  $S_2$  abzuschneiden, daher wird die komplementäre Fehlerfunktion über folgende Beziehung genähert:

$$\operatorname{erfc}(x) = \operatorname{tp}(x) \exp(-x^2) + \epsilon(x) \quad |\epsilon(x)| \leq 1.5 \cdot 10^7, \quad (2.36)$$

wobei  $\operatorname{tp}(x)$  ein Polynom fünften Grades in  $x$  ist (Koeffizienten aus [89]).

Die Summe  $S_1$  konvergiert umso schneller, je größer der Parameter  $\alpha$  gewählt ist. Damit legt  $\alpha$  den Cutoff–Radius  $r_c^{\text{ew}}$  fest, bei dem  $S_1$  abgeschnitten werden kann. Unter Verwendung der Minimum–Image–Convention muß beachtet werden, daß  $r_c^{\text{ew}}$  kleiner als die halbe Boxlänge ist (siehe dazu Kapitel 2.1.1.5).

Während im Ortsraum der Rechenaufwand mit steigendem  $\alpha$  abnimmt, nimmt die Zahl der Gittervektoren  $\vec{m}$  mit  $|\vec{m}| < m_c$  (mit  $m_c$  den Gitterzahlcutoff), zu, um eine gewünschte Genauigkeit zu erreichen. Das Ziel ist daher, die Parameter  $\alpha$  und  $m_c$  derart geschickt zu wählen, daß die Genauigkeit im gewünschten Rahmen bleibt, der Rechenaufwand jedoch minimal

wird. Unter vorgegebenen Parametern  $\alpha$  und  $m_c$  erhält man nun die Form des Coulombpotentials zur numerischen Berechnung:

$$\begin{aligned}
 V_C = & \sum_i \sum_{\substack{j > i \\ r_{ij} < r_c^{\text{ew}}}} \frac{q_i q_j e^2}{r_{ij}} \text{erfc}(\alpha r_{ij}) - \frac{\alpha}{\sqrt{\pi}} \sum_{j=1}^N q_j^2 e^2 \\
 & + \frac{1}{2\pi L} \sum_{\substack{\vec{m} \in \mathbb{Z}^3 \\ 0 < m < m_c}} \frac{\exp\left(-\left(\frac{\pi m}{\alpha L}\right)\right)}{m^2} \left| \sum_{j=1}^N q_j e \exp\left(i \frac{2\pi}{L} \vec{r}_j \cdot \vec{m}\right) \right|^2 \quad (2.37)
 \end{aligned}$$

Der Gradient des Potentials ergibt die Coulombkraft  $\vec{F}_{c,i}$  auf das Teilchen  $i$ :

$$\begin{aligned}
 \vec{F}_{c,i} = & -\frac{\partial V_C}{\partial \vec{r}_i} \\
 = & \sum_{j=1}^N q_i q_j \left\{ \frac{2\alpha}{\sqrt{\pi}} \exp\left(-(\alpha r_{ij})^2\right) + \frac{\text{erfc}(\alpha r_{ij})}{r_{ij}} \right\} \frac{\vec{r}_{ij}}{r_{ij}^2} - \\
 & -\frac{2}{L^2} \sum_{\substack{\vec{m} \in \mathbb{Z}^3 \\ 0 < m < m_c}} q_i e \vec{m} \frac{\exp\left(-\left(\frac{\pi m}{\alpha L}\right)^2\right)}{m^2} \times \\
 & \times \text{Im} \left( \exp\left(-i \frac{2\pi}{L} \vec{r}_i \cdot \vec{m}\right) \sum_j q_j e \exp\left(i \frac{2\pi}{L} \vec{r}_j \cdot \vec{m}\right) \right). \quad (2.38)
 \end{aligned}$$

Mit  $\lim_{z \rightarrow \infty} \frac{\text{erfc}(z)}{z} = \frac{\exp(-z^2)}{z^2}$  folgt aus Gleichung (2.37), daß sich der Fehler durch das Abschneiden der Ortsraumsumme wie  $\frac{\exp(-s^2)}{s^2}$  mit  $s = \alpha r_c^{\text{ew}}$  verhält. Der entsprechende Fehler im reziproken Raum ist  $\frac{\exp(-\hat{s}^2)}{\hat{s}^2}$  mit  $\hat{s} = \frac{\pi m_c}{\alpha L}$ . Dadurch erhält man die Bedingungen:

$$\begin{aligned}
 r_{ij} < r_c^{\text{ew}} &= \frac{s}{\alpha} \\
 m < m_c &= \frac{\hat{s} \alpha L}{\pi}.
 \end{aligned}$$

Bewährt hat sich eine Kombination aus  $s = 3$  und  $r_c^{\text{ew}} = 10.15 \text{ \AA}$ . Dadurch lassen sich  $\alpha$  und  $m_c$  bestimmen (der Cutoff-Fehler liegt bei  $\approx 10^{-5}$ ).

Zur Abschätzung des Rechenzeitaufwands nimmt man eine homogene Verteilung der Teilchen an. Die Rechenzeit zur Bestimmung des Realraumanteils der Ewaldsumme (Kraft und Potential) skaliert dann mit  $N^2$  [88]. Die Zeit zur Berechnung des Imaginärteils ist proportional zu  $N$ . Durch geeignete Wahl der Parameter (s.o.) kann man die Berechnung von Gleichung (2.37) und Gleichung (2.38) so skaliert werden, daß die Rechenzeit effektiv  $\propto N^{3/2}$  skaliert.

$\alpha - \beta$	$A_{\alpha\beta}$ [eV]	$B_{\alpha\beta}$ [ $\text{\AA}^{-1}$ ]	$C_{\alpha\beta}$ [ $\text{eV}\text{\AA}^6$ ]
Si-Si	872356916	15.2207	23.30
Si-Li	239064	9.4286	10.93
Si-Na	82097	8.5135	–
Si-K	146920	7.4878	–
Si-O	10722	2.8464	70.73
Li-Li	6712	6.8297	5.12
Li-Na	4010	6.3363	–
Li-K	8507	5.7501	–
Li-O	2388	4.0182	33.17
Na-Na	2580	5.9095	–
Na-K	5462	5.3964	–
Na-O	1870	3.8422	–
K-K	10327	4.9652	–
K-O	3227	3.6185	–/5.0
O-O	1757	2.8464	214.74

**Tabelle 2.3.:** Potentialparameter für die Alkalisilikat–Wechselwirkung nach Habasaki and Okada [71]. Der farbig markierte Parameter  $C_{\text{KO}} = 5.0 \text{ eV}\text{\AA}^6$  stellt eine Modifikation des Potentials dar (siehe Infokasten auf Seite 34).

Der Algorithmus zur Berechnung von Kraft und Potential kann noch weiter optimiert werden, so daß die Summe der reziproken Gittervektoren  $\vec{m}$  sich auf die Hälfte reduziert:

$$\sum_{\vec{m} \in \mathbb{Z}^3} = c \cdot \sum_{m_x=0}^{m_{\max}} \sum_{\substack{m_y = -m_{\max} \\ m \leq m_c = \sqrt{m_{\max}^2 + 2}}} \sum_{m_z=-m_{\max}}^{m_{\max}} \quad \text{für } c = \begin{cases} 1 & m_x=0 \\ 2 & \text{sonst} \end{cases} \quad (2.39)$$

Die Bedingung  $m \leq m_c = \sqrt{m_{\max}^2 + 2}$  stellt sicher, daß nicht nur Gittervektoren innerhalb der Cutoff–Kugel berücksichtigt werden, sondern auch diejenigen Vektoren, die auf den Raumdiagonalen in  $\mathbb{Z}^3$  liegen.

Zusätzlich wird der konfigurationsabhängige Teil der reziproken Gittersumme in Potential und Kraft  $\left[ \exp\left(-\left(\frac{\pi m}{\alpha L}\right)^2\right) \right] / m^2$  zu Beginn jedes Simulationslaufs tabelliert.

## 2.1.2. Mikroskopisches Modell

### 2.1.2.1. Potential

Das Potential entscheidet maßgeblich über die Güte einer MD–Simulation. Erst durch ein entsprechend gut modelliertes Potential können Materialeigenschaften nachvollzogen, vorhergesagt und auf einer mikroskopischen Ebene verstanden werden.



### Probleme der Potentialfindung I: Von BKS bis Habasaki.

Wie eingangs in Kapitel 2.1.2.1 erwähnt, ist die Wahl eines geeigneten Potentials von entscheidender Bedeutung für die Güte der Simulation. Daher stand am Anfang dieser Arbeit der Versuch, ein bekanntes und weithin getestetes Silikatpotential [90, 15, 91, 14, 92, 93, 94, 95, 96, 97, 98, 99, 100] abzuwandeln: Das sogenannte *BKS-Potential* von van Beest et al. [101]. Von der Form her ist es identisch zu Gleichung (2.41), es unterscheidet sich nur in den Parametern.

1991 wurde dieses von Kramer et al. [102] auch für eine Alkalisorte, für Natrium, erweitert. Da mit der von Kramer et al. vorgeschlagenen Ladung für Natrium jedoch keine Ladungsneutralität im System besteht, wurde dem Potential noch ein effektives Potential hinzugefügt, daß sie Ladungsneutralität gewährleistet (vgl. [6, 17]). Leider fehlte hier für die Bearbeitung des MAE die zweite Alkali-Komponente. Daher wurden zu Anfang dieser Arbeit Anstrengungen unternommen, aus der Natriumkomponente von Kramer et al. eine zweite Alkalikomponente zu „konstruieren“. Folgende Ansätze wurden dabei verfolgt:

- Vom chemischen Standpunkt her kann man die Atome einer Gruppe als recht ähnlich ansehen. Sie unterscheiden sich im wesentlichen im Radius. Daher wurde versucht, durch Variieren der Parameter  $A_{\alpha\beta}$  und  $B_{\alpha\beta}$  das Minimum zu größeren Radien zu verschieben (wobei das Energieminimum automatisch mit angehoben wird). Das durch das resultierende Potential beschriebene abgewandelte Alkaliion wird folgend als Na' bezeichnet.  
Dieser Versuch scheiterte auf recht „interessante“ Art. Anstatt eines MAE wie erwartet, erhielt man in einem Plot der Diffusionskonstante über Zusammensetzung folgendes Ergebnis: Ausgehend von einem binären Natriumdisilikat, mit Natrium so wie Kramer et al. es definierten, fällt die Diffusionskonstante hin zu einem ternären System mit Na' hin ab. Aber statt irgendwann, wenn der Anteil von Na' größer als Na ist, anzuwachsen, fällt die Diffusionskonstante auch hin zu binären Na'disilikat immer weiter ab.
- Es wurden zwei ungleichnamige Natriumsorten in das System eingeführt, deren Potential identisch ist, die untereinander aber eine abstoßende Wechselwirkung zeigten.
- Die Masse der Natriumkomponente wurden variiert.

All jene Ansätze führten nicht zum Ziel, zunächst einmal den MAE zu beobachten. Daher wurde das in Kapitel 2.1.2.1 beschriebene Potential von Habasaki and Okada verwendet.

Daß das allerdings auch nicht ganz unproblematisch ist, ist im Infokasten auf Seite 35 nachzulesen.

Gibt es aber auf der mikroskopischen Ebene noch kein hinreichendes Verständnis über die Wechselwirkung unter den Teilchen, um ein Potential für klassische Simulationen zu formulieren, so bieten sich quantenmechanische Ab-Initio-Methoden an. Eine von diesen ist z.B. die Car-Parinello-Molekulardynamik (CPMD) [103]. Diese Methode berücksichtigt die elektronischen Freiheitsgrade im Rahmen einer Dichtefunktionaltheorie und erfordert daher nicht wie die klassische Molekulardynamik die Verwendung einer (empirischen) Potentialfunktion, die ja gerade die elektronischen Freiheitsgrade effektiv beschreibt. In der CPMD werden gewissermaßen die Wechselwirkungen zwischen den Atomen „on the fly“ berechnet. Das resultiert bei der praktischen Umsetzung von Simulationen in viel geringeren Teilchenzahlen (in der Größenordnung von 100) und viel kürzeren Simulationszeiten (in der Größenordnung von einigen ps), da der Rechenaufwand um Größenordnungen höher ist, als bei klassischen Simulationen.

Eine relativ gut mit der experimentellen Realität übereinstimmende und zugleich effiziente Methode zur Simulation von mikroskopischen Systemen erhält man, indem man mit kleinen Systemen Ab-Initio-Simulationen durchführt, und die daraus erhaltenen Ergebnisse verwendet, ein klassisches Potential zu optimieren, das möglichst viele Merkmale des Ab-Initio-Systems aufweist.

Dieser Ansatz wurde auch bei der Optimierung des in dieser Arbeit verwendeten Potentials

	q
Si	2.4
O	$-\frac{N_{\text{Si}}q_{\text{Si}} + \sum_i N_{X_i}q_{X_i}}{N_{\text{O}}}$
Li	0.87
Na	0.88
K	0.85

**Tabelle 2.4.:** Effektive Ladungen der Teilchensorten nach Habasaki and Okada [71]. Wichtig dabei ist die Auswahl der Sauerstoffladung derart, daß das Gesamtsystem Ladungsneutral bleibt (siehe Text).  $N_x$  ist die Anzahl einer Teilchensorte im System,  $q_x$  deren Ladung.  $X_i$  steht für jeweils eine Alkali-Teilchensorte.

verfolgt. Dabei wurde ein Modell von Habasaki and Okada [71] verwendet, in welchem die Parameter der Alkali-Wechselwirkungen anhand von quantenmechanischen Rechnungen an  $((e^+ - O - X - O - e^+)^{-1})$ -Clustern optimiert wurden ( $X \in \{\text{Li, Na, K}\}$ ). Grundlage dafür war ein Potential für  $\text{SiO}_2$  von Tsuneyuki et al. [70], welches ebenfalls durch Ab-Initio-Rechnungen optimiert wurde.

Das Gesamtpotential eines Teilchens ergibt sich durch Summation über alle Paarpotentiale  $\phi_{ij}$

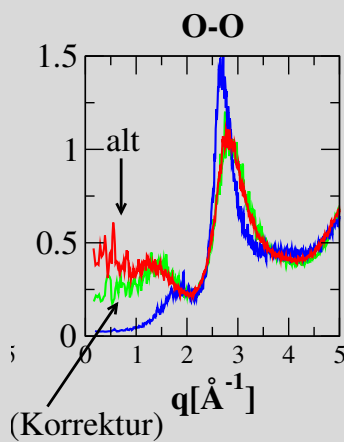
$$V(\{r_i\}) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \phi_{ij}(r_{ij}) \quad r_{ij} = |\vec{r}_i - \vec{r}_j| \quad (2.40)$$

mit der Gesamtteilchenzahl  $N$  und den Teilchenpositionen  $r_i$ ,  $i = 1, \dots, N$ . Das optimierte klassische Paarpotential ist vom Gilbert-Ida Typ mit einem  $r^{-6}$ -Term und hat folgende Form,

$$\phi_{\alpha\beta}(r) = \underbrace{\frac{q_\alpha q_\beta e^2}{r}}_{(1)} + \underbrace{A_{\alpha\beta} \exp(-B_{\alpha\beta} r) - \frac{C_{\alpha\beta}}{r^6}}_{(2)}, \quad (2.41)$$

besteht also aus einem Coulombterm (1) mit effektiven Partialladungen (siehe Tabelle 2.4) und der Elementarladung  $e$ .  $r$  ist der Abstand zwischen den Atomen der Sorte  $\alpha$  und  $\beta$ . Der Buckinghamanteil (2) besteht aus einem abfallendem Exponentialterm und einem van der Waals-Term, der hier nicht explizit die Dipol-Dipol-WW beschreibt, sondern nur einen Freiheitsgrad für die Optimierung des Potentials liefert. Der Buckinghamanteil beschreibt den kovalenten Teil der Bindung.

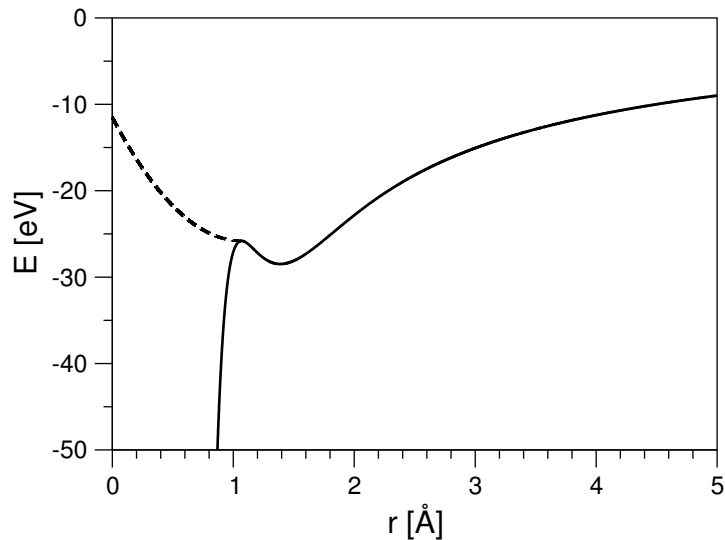
### Probleme der Potentialfindung II: Habasaki und Verbesserungen.



Das Potential von Habasaki and Okada [71] erscheint anfangs sehr vielversprechend, da es auch ein vielverwendetes [71, 104, 30, 105, 22, 106], in Teilen gut untersuchtes Potential darstellt.

In Teilen deshalb, da die Potentiale für Natrium- und Kaliumsilikate in den partiellen Strukturfaktoren der Alkalikomponenten Entmischungstendenzen aufweisen (relativ große Amplitude für  $q \rightarrow 0$ ). Um dem hier vorzubeugen, wurde ein zusätzlicher Parameter gesetzt, der zu einer stärkeren K - O-Bindung führt (siehe Tabelle 2.3). Dadurch senkte sich die Kompressibilität, ohne daß sich an der sonstigen Struktur in intermediären oder kurzen Längenskalen etwas änderte.

Zu sehen ist dies links. Blau ist  $S_{OO}(q)$  für ein Lithiumdisilikat. Rot und grün sind  $S_{OO}(q)$  für Kaliumdisilikat, wobei rot dem Potential nach Habasaki and Okada entspricht und grün der Strukturfaktor mit der in dieser Arbeit vorgenommenen Anpassung.



**Abbildung 2.6.:** Beispiel für die Erweiterung des Potentials um einen repulsiven, quadratischen Term. Hier wird das Potential für die Si–O–Wechselwirkung (durchgezogene Linie) in KS2 gezeigt. Um unphysikalisch geringe Abstände aufgrund des stark attraktiven Potentialanteils für  $r < 1.07 \text{ \AA}$  zu vermeiden, wird ein repulsives Potential angefügt (gerstrichelte Linie).

Die Parameter  $A_{\alpha\beta}$ ,  $B_{\alpha\beta}$  und  $C_{\alpha\beta}$  kann man Tabelle 2.3 entnehmen. Dabei sind die effektiven Ladungen so zu wählen, wie in Tabelle 2.4 angegeben. Wichtig ist hierbei, daß die Sauerstoffladung so eingestellt wird, daß das System ladungsneutral ist. Ist dies nicht der Fall, so konvergiert die Ewald–Summe (vgl. Kapitel 2.1.1.7) nicht mehr.

Zudem erlaubt dieses Vorgehen der Anpassung der Sauerstoffladung eine Art Ladungstransfer zwischen X und O. Das ergab in [71] deutlich bessere Ergebnisse in Dichte und Kristallstruktur, als ein Modell ohne Ladungstransfer.

Für die Umsetzung in der Simulation ist allerdings noch ein Punkt zu beachten. Bei einigen Wechselwirkungen wird für kleine Radien  $r$  der Van–der–Waals–Term stark dominant, was dazu führt, daß die Wechselwirkungen für kleine  $r$  unter ca.  $1 \text{ \AA}$  attraktiv wird. Das ist zwar ein Bereich, den die Teilchen nur bei sehr hohen Temperaturen häufiger erreichen, verhindert bei Auftreten allerdings die Simulationsfortsetzung, da die Teilchen dann in dieser attraktiven Wechselwirkung gefangen bleiben und die Energie divergiert. Daher wurde das Problem umgangen, indem man in diesem Bereich ein abstoßendes quadratisches Potential angefügt hat (Abbildung 2.6).

### 2.1.2.2. Vergleich mit dem Experiment

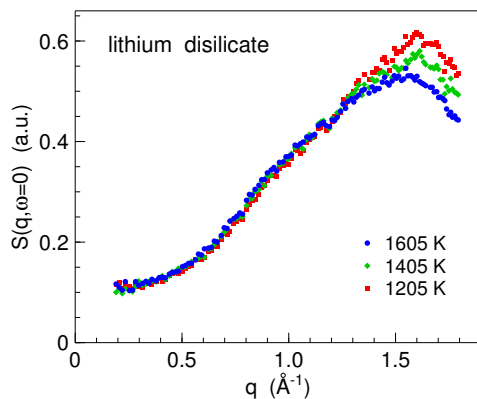
Letztendlich ist die Güte eines Potentials nur durch Vergleiche mit experimentell oder per Ab–Initio–Rechnungen erlangten Größen und Effekten möglich. Natürlich wurden von Habasaki and Okada [71] viele Parameter bei der Potentialoptimierung überprüft, allerdings nicht für die Systeme, die in dieser Arbeit untersucht werden, sondern für Kristalle.

Zunächst ist eine Möglichkeit, kleine Längenskalen in Simulation und Experiment zu ver-

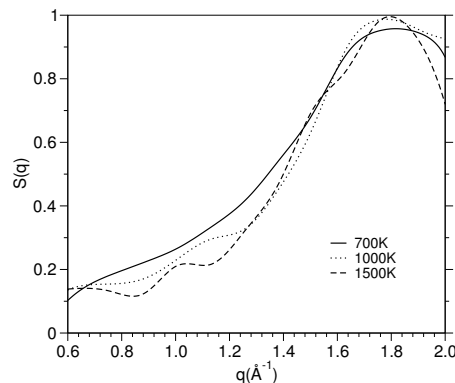
LS2	Exp. [Å]	Sim. [Å]
Si–Si	3.13	3.15
Si–O	1.61	1.62
Li–O	2.08	2.02
O–O	2.66	2.65

KS2	Exp. [Å]	Sim. [Å]
Si–Si	3.23	3.14
Si–O	1.62	1.63
K–O	2.66	2.67
O–O	2.66	2.67

**Tabelle 2.5.:** Maxima-Positionen des ersten Peaks der partiellen Paarkorrelationsfunktionen für LS2 und KS2, aus Simulation (Sim., bei 1000 K) und Experimenten (Exp., ca. 1150 K). (Quelle: Mazurin et al. [107]).



**Abbildung 2.7.:** Elastische Neutronenstreu- und Strukturformfaktoren von Kargl [35].



**Abbildung 2.8.:** Elastische Strukturformfaktoren ( $S(q)$  ist hier  $S(q, \omega = 0)$ ) aus Simulationen zu Lithiumdisilikat.

gleichem, die partielle Paarkorrelationsfunktion  $g_{\alpha\beta}(r)$ . Wie man in Tabelle 2.5 sieht, sind die größten Abweichungen  $\leq 3\%$ . Die meisten Peakpositionen zeigen eine sehr gute Übereinstimmung. Dies zeigt, daß auf kleinen Längenskalen das verwendete Potentialmodell realistisch ist.

Für größere, intermediäre Längenskalen, bietet sich der Vergleich der elastischen Strukturformfaktoren<sup>10</sup> an. Von Kargl [35] stammen diesbezüglich neue Daten, die in Abbildung 2.7 mit den Resultaten aus der Simulation (Abbildung 2.8) verglichen werden. Im Vergleich zu den experimentellen Daten in Abbildung 2.7 sieht man hier eine Verschiebung zu höheren  $q$ -Werten. Obwohl nicht so ausgeprägt wie im Experiment, ist auch ein mit steigender Temperatur zurückgehender Peak um  $q = 1.0 \text{ Å}^{-1}$  herum zu erkennen, analog zur Schulter bei ca.  $0.9 \text{ Å}^{-1}$  in Abbildung 2.7. Für ein klassisches Paarpotential, wie es hier verwendet wurde, ist die qualitative Übereinstimmung recht gut.

### 2.1.3. Parallelisierung und praktische Durchführung

Da der Rechenaufwand für die untersuchten Systeme und Systemgrößen für einzelne Prozessoren zu langwierig ist, mußte das Programm durch Parallelisierung des Computercodes beschleunigt werden. Dazu wurde das *Message Passing Interface* (MPI) [108] verwendet.

<sup>10</sup>vgl. Kapitel 2.2.1.7.

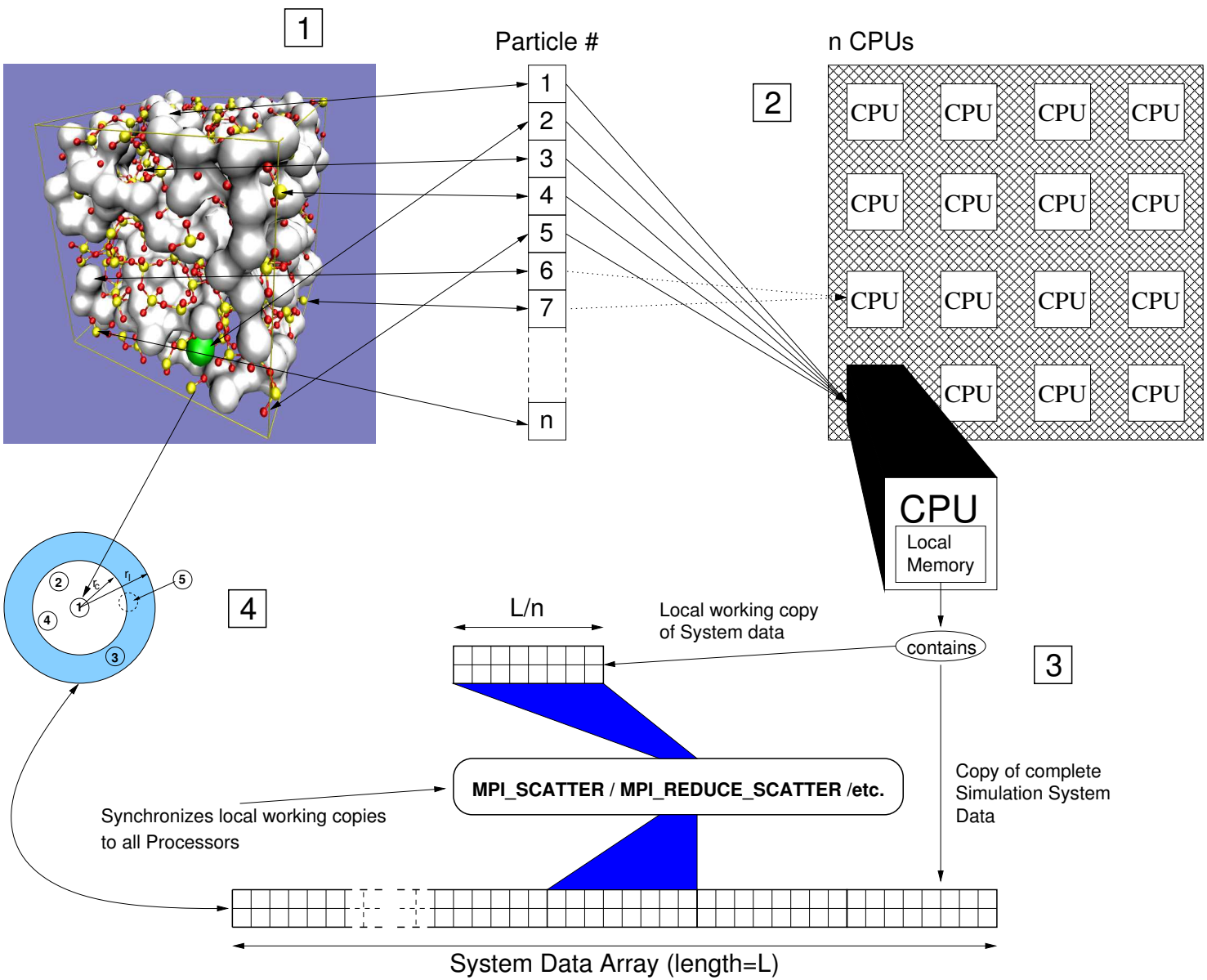


Abbildung 2.9.: Diagramm der Funktionsweise der Parallelisierung des Simulationsprogrammcodes.

Dieser Standard ist auf Parallelrechnern und entsprechend ausgebauten Rechenclustern weltweit verbreitet und seine Implementationen bieten die Möglichkeit, zwischen den Prozessoren, unabhängig von der zugrundeliegenden Rechnerarchitektur, effizient zu kommunizieren.

Hier soll nicht auf die Einzelheiten der Parallelisierung eingegangen werden. Für weitere Details wird auf die Doktorarbeit von Horbach [6] verwiesen, in der Details zur Parallelisierung eines Vorläuferprogramms besprochen werden. An den Prinzipien, die dort diskutiert werden, hat sich in dieser Arbeit bezüglich MPI nichts geändert.

Der Ansatz der Parallelisierung soll hier mittels Abbildung 2.9 veranschaulicht werden. Die folgende Aufzählung bezieht sich auf die Nummern in rechteckigen Kästen in Abbildung 2.9:

1. Ausgangspunkt der Simulation ist eine Konfiguration von in dieser Arbeit zumeist 8064 Teilchen. Das ergibt Simulationsboxgrößen von ca.  $(50 \text{ \AA})^3$ . Um den Rechenaufwand möglichst gleichmäßig auf die zur Verfügung stehenden Prozessoren zu verteilen, werden den Prozessoren nicht Raumbereiche mit Teilchen zugeteilt (z.B. *Boxing*). Stattdessen werden jedem Prozessor die gleiche Anzahl Teilchen zugewiesen, die ihre eigene Nachbarschaftsliste mitführen (vgl. Punkt 4). Daher gilt als Voraussetzung für das Programm, daß  $\#(\text{Teilchen}) = n \cdot \#(\text{CPU})$  mit  $n \in \mathbb{N}$ .
2. Jeder Prozessor hat einen lokalen Speicher. In diesem Speicher enthalten sind einerseits lokale Variablen und lokale Ausschnitte von großen Datenblöcken, z.B. die Positionsdaten der dem Prozessor zugeteilten Teilchen. Andererseits enthalten die lokalen Speicher der Prozessoren auch Kopien der Gesamtdaten, z.B. auch eine Kopie der Positionsdaten.
3. Mittels Befehlen, die durch die MPI-Schnittstelle bereitgestellt werden, können die lokalen Kopien von Daten und Variablen mit den globalen Daten und Variablen synchronisiert werden. Z.B. kann eine Größe wie die Systemenergie durch einen MPI-Befehl von lokalen Teilenergien aufaddiert werden. Oder lokale Positionsdaten werden, wie in Abbildung 2.9 durch die dunkelblauen Strahlen angedeutet, mittels eines MPI-Befehls auf allen lokalen Kopien der globalen Positionen korrekt gespiegelt.
4. Dadurch wird sichergestellt, daß die Wechselwirkungen mit den Nachbarn korrekt berechnet werden, ohne daß jedes Teilchen auf die Antwort eines anderen Prozessors warten muß, wo sich jetzt das Teilchen  $n$  befindet. (Weitere Details zur Nachbarschaftslistenberechnung finden sich in Kapitel 2.1.1.6).

### 2.1.4. Übersicht simulierter Systeme

Hier soll nun in Tabelle 2.6 ein Kurzüberblick über alle simulierten Systeme gegeben werden.

Zunächst wurden die binären Systeme **LS2** und **KS2** simuliert. Diese Abkürzungen stehen für  $(\text{Li}_2\text{O}) \cdot (2 \cdot \text{SiO}_2)$ , also Lithiumdisilikat mit 33.3 mol% Lithium und für  $(\text{K}_2\text{O}) \cdot (2 \cdot \text{SiO}_2)$ , also Kaliumdisilikat mit 33.3 mol% Kalium. Das System besteht aus 1792 Si, 1792 Li bzw. K und 4480 O.

Temperatur [K]	Systeme	$\sim t_{\text{equi}}$ [ns]	$\sim t_{\text{sample}}$ [ns]
700	LS2/KS2/LKS2	11	11
1000	LS2/KS2/LKS2	78.6	7.86
1000	LS2+1K	7.86	7.86
1250	LS2/KS2/LKS2	11	11
1500	LS2/KS2/LKS2	11	11
1850	LS2/KS2/LKS2	11	11
2000	LS2/KS2/LKS2	3.14	3.14
2000	LS2+1K	1.57	1.57
2300	LS2/KS2/LKS2	4.72	4.72
3000	LS2/KS2/LKS2	3.14	$\leq 3.14$
4000	LS2/KS2/LKS2	1.57	1.57
4000	LS2+1K	0.0786	0.078

**Tabelle 2.6.:** Übersicht über Simulationsdetails der für diese Arbeit simulierten Systeme.

Um den kollektiven MAE zu untersuchen, wurde **LKS2** simuliert. LKS2 steht für  $(0.5 \cdot \text{Li}_2\text{O}) \cdot (0.5 \cdot \text{K}_2\text{O}) \cdot (2 \cdot \text{SiO}_2)$ , also ternäres Alkalidisilikat mit 33.3 mol% Alkaliionenanteil, gleichmäßig aufgeteilt in Lithium und Kalium. Das System besteht aus 1792 Si, 896 Li, 896 K und 4480 O.

LKS2 stellt jedoch nicht den Extremfall des MAE dar<sup>11</sup>, daher wurde noch **LS2+1K** simuliert. LS2+1K steht für ein LS2-System mit 1152 Teilchen, bei der ein Li-Ion gegen ein K-Ion getauscht wurde (was etwas irreführend ist, da im Namen „+1K“ steht). Das System wird mit 256 Si, 255 Li, 1 K und 640 O simuliert. Für LS2+1K wurden bei 1000 K 20 unabhängige Läufe gemacht und bei den höheren Temperaturen jeweils 8, um eine relevante Menge an Messungen zu erhalten.

Der Simulationsablauf gestaltete sich immer nach dem gleichen Schema. Für hohe Temperaturen (3000 – 4000 K) werden Ausgangskonfigurationen, wie sie in Kapitel 2.1.1.3 beschrieben wurden, verwendet. Alle niedrigeren Temperaturen haben Endkonfigurationen aus Simulationsläufen höherer Temperatur als Anfangskonfiguration. Die Anfangskonfiguration wird noch die Zeitdauer  $t_{\text{equi}}$  im  $NPT$ -Ensemble bei der Zieltemperatur (der gewünschten Temperatur des Systems) äquilibriert. Dann wird im  $NPH$ -Ensemble für ca. 16 ps nachäquilibriert, um das Volumen ohne Einfluß des Thermostaten zu äquilibrieren. Auf Basis des Mittelwerts  $l_{\text{mean}}$  der Boxlänge des  $NPH$ -Simulationslaufs werden verschiedene Variationen der Boxlänge  $\{l_{\text{mean}} - 0.1, \dots, l_{\text{mean}} + 0.1\}$  für ca. 1.6 ps simuliert, um eine Boxlänge zu bestimmen, bei der der mittlere Druck  $< 10^{-2}$  GPa ist. Mit dieser Boxlänge wird dann der  $NVE$ -Simulationslauf durchgeführt, der  $t_{\text{sample}}$  lange dauert. Aus diesem Simulationslauf stammen die Daten, die für die Auswertung verwendet werden.

<sup>11</sup>vgl. Kapitel 4.

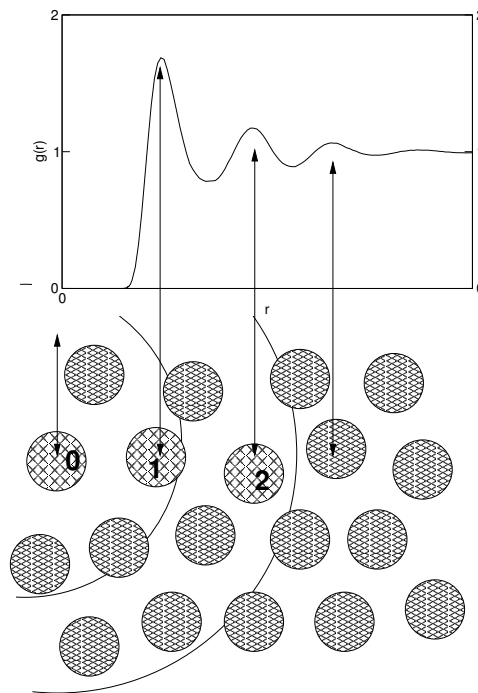


Abbildung 2.10.: Illustration zur Interpretation der Paarkorrelationsfunktion.

## 2.2. Analysemethoden

### 2.2.1. Strukturelle Größen

Mittels der im folgenden vorgestellten Größen werden in dieser Arbeit statische Struktureigenschaften des Alkalisilikatmodells untersucht. Angefangen wird dabei mit den Paarkorrelationen, mit denen man die direkte Umgebung um die Teilchen gut auf ihre Nachbarschaft mit anderen Teilchen hin untersuchen kann. Mittels der Paarkorrelation ist man dann in der Lage, Koordinationsfunktionen zu definieren, die Aufschluß über die chemische Umgebung eines Teilchens liefern, also mit welchen Teilchen z.B. Bindungen bestehen. Sind diese Bindungen erst einmal identifiziert, kann mittels der Bond–Bond Winkelverteilung untersucht werden, in welchen Winkelverteilungen sich die Bindungen einstellen. Unter Kenntnis der Bindung können dann auch geschlossene SiO Ringe gesucht werden, um Ringlängenverteilungen zu betrachten. Zur Untersuchung von intermediären bis großen Längenskalen eignen sich die Strukturfaktoren, die in der Form vom elastischen– bzw. „Neutronenstreu“–Strukturfaktor dann auch mit experimentellen Ergebnissen verglichen werden können.

#### 2.2.1.1. Die partielle Paarkorrelationsfunktion $g_{\alpha\beta}(r)$

Die partiellen Paarkorrelationsfunktionen  $g_{\alpha\beta}(r)$  sind besonders wichtig, um detaillierte Aussagen über die lokale Struktur zu machen. Die Idee dahinter ist, eine Größe zu definieren, die das Maß an struktureller Korrelationen mit den Nachbarteilchen in Abhängigkeit der Entfernung  $r$  vom markierten Teilchen angibt [109]. Voraussetzung dafür ist, daß es sich um ein



isotropes bzw. ungeordnetes System handelt, da man sonst richtungsabhängige Korrelationen erhält (z.B. im Kristall).

Die Paarkorrelationsfunktion ist dabei eine einfache Größe: Sie ist einfach das Verhältnis zwischen der mittleren Anzahldichte  $\rho(r)$  in einem Abstand  $r$  von jedem vorgegebenen Teilchen<sup>12</sup> und der Dichte eines idealen Gases im Abstand  $r$  mit der selben Gesamtdichte. Dabei gilt die Beziehung (vgl. [109])

$$g_{\alpha\beta}(r) \underset{r \rightarrow \infty}{\longrightarrow} 1 - \frac{1}{N}.$$

Für Systeme mit  $N \gg 1$  besagt das Resultat  $g_{\alpha\beta}(r) = 1$ , daß die Teilchen zufällig verteilt vorliegen; Abweichungen von  $g_{\alpha\beta}(r) = 1$  signalisieren das Vorhandensein von Korrelationen.

Die Definition der Berechnung lautet wie folgt:

$$g_{\alpha\beta}(r) = \mathcal{N}_{\alpha\beta} \left\langle \sum_{i=1}^{N_\alpha} \sum_{j=1}^{N_\beta} \frac{1}{4\pi r^2} \delta(r - |\vec{r}_i - \vec{r}_j|) \right\rangle \quad \alpha, \beta \in \{\text{Si, O, Li, K}\} \quad (2.42)$$

mit den Normierungskonstanten

$$\mathcal{N}_{\alpha\beta} = \begin{cases} \frac{N}{\rho N_\alpha (N_\alpha - 1)} & \alpha = \beta \\ \frac{N}{\rho N_\alpha N_\beta} & \alpha \neq \beta \end{cases} \quad (2.43)$$

Dabei ist  $N_\alpha$  die Anzahl der Teilchen der Sorte  $\alpha$ ,  $N$  die Gesamtzahl der Teilchen,  $|\vec{r}_i - \vec{r}_j|$  der Abstandsbetrag der Teilchen  $i$  und  $j$  und  $\rho$  die Teilchenzahldichte im System.  $4\pi r^2 g_{\alpha\beta}(r)$  ist proportional zur Wahrscheinlichkeit, ein Teilchen der Sorte  $\beta$  im Abstand  $r$  zu einem Teilchen der Sorte  $\alpha$  zu finden (siehe Abbildung 2.10).

Oftmals spricht man im Zusammenhang mit Paarkorrelationsfunktionen in Glas oder Flüssigkeiten von sogenannten *Nächsten-Nachbar-Schalen*. Die Definition wird durch Abbildung 2.10 veranschaulicht. Den ersten Peak nennt man „first next neighbor peak“, da er den Abstand zu der ersten nächsten Nachbar-„Schale“ angibt, wie zwischen Teilchen 0 und 1 schematisch dargelegt. Die darauffolgenden Peaks korrespondieren entsprechend mit den „Schalen“ höherer Ordnung (mit Kreisabschnitten angedeutet).

### 2.2.1.2. Die Koordinationszahlverteilung

Mithilfe der gerade definierten partiellen Paarkorrelationsfunktion kann man auch die Koordinationszahlverteilung  $P_{\alpha\beta}$  bestimmen. Wie gerade erwähnt ist  $4\pi r^2 g_{\alpha\beta}(r)$  proportional zur Wahrscheinlichkeit, ein Teilchen der Sorte  $\beta$  im Abstand  $r$  zu einem Teilchen der Sorte  $\alpha$  zu finden. Der Proportionalitätsfaktor dafür ist die mittlere Anzahldichte  $\rho_\alpha = \frac{N_\alpha}{V}$  [110]. Damit erhält man dann die Anzahl Teilchen der Sorte  $\beta$  in der Kugel mit dem Radius  $r$  um ein Teilchen der Sorte  $\alpha$  als

<sup>12</sup>Zur Vereinfachung nimmt man hier identische Teilchen an.

$$\begin{aligned}
 P_{\alpha\beta}(r) &= 4\pi\rho_{\beta} \int_0^r dr r^2 g_{\alpha\beta}(r) \\
 &= \frac{1}{N_{\alpha}} \sum_{i=1}^{N_{\alpha}} \sum_{j=1}^{N_{\beta}} \Theta(-(|\vec{r}_i - \vec{r}_j| - r))
 \end{aligned}$$

mit  $\Theta(x)$  als der Heavyside-Funktion

$$\Theta(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases} .$$

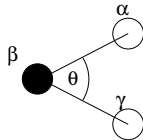
Berechnet man  $P_{\alpha\beta}(r)$  für  $r = r_{\alpha\beta,\min 1}$ , wobei  $r_{\alpha\beta,\min 1}$  die Position des ersten Minimum der partiellen Paarverteilungsfunktion  $g_{\alpha\beta}(r)$  ist, so erhält man die Koordinationszahlverteilung

$$P_{\alpha\beta} = P_{\alpha\beta}(r_{\alpha\beta,\min 1}) = \frac{1}{N_{\alpha}} \sum_{i=1}^{N_{\alpha}} \sum_{j=1}^{N_{\beta}} \Theta(|\vec{r}_i - \vec{r}_j| - r_{\alpha\beta,\min 1}) .$$

Das Resultat ist die mittlere Anzahl der Teilchen der Sorte  $\beta$ , die sich in der ersten Schale (vgl. Abbildung 2.10) um ein Teilchen der Sorte  $\alpha$  befindet.

### 2.2.1.3. Bond-Bond-Winkelverteilungen

Die *Bond-Bond-Winkelverteilung*  $P_{\alpha\beta\gamma}(\theta)$  gibt die Wahrscheinlichkeit an, den Winkel  $\theta$  zwischen 3 Teilchen der Sequenz  $\alpha\beta\gamma$  vorzufinden, wobei  $\alpha$ ,  $\beta$  und  $\gamma$  verschiedene Teilchen sind und die  $\beta$ -Teilchen sich innerhalb des ersten  $\alpha - \beta$  und  $\beta - \gamma$  Nächste-Nachbar Bereich befinden, d.h. der Betrag des Abstandsvektors ist kleiner als die Position des ersten Minimums der Paarkorrelationsfunktion  $g_{\alpha\beta}(r)$ . Anschaulich ist dies in Abbildung 2.11 dargestellt.



**Abbildung 2.11.:** Graphische Repräsentation der Definition des Bond-Bond Winkels der Teilchen  $\alpha - \beta - \gamma$ . Bestimmt wird der Winkel  $\theta$  auf der Ebene, die durch die 3 Teilchen definiert wird.

### 2.2.1.4. Ringlängenverteilungen

Die *Ringlängenverteilung* ist eine Größe, die besonders dazu geeignet ist, netzwerkbildende Systeme wie z.B. amorphes  $\text{SiO}_2$  zu untersuchen. Das Ausbilden von Ringen ist darin ein typisches Verhalten und gibt Aufschluß über Struktur und Anzahl mehrerer verbundener  $\text{SiO}_4$ -Tetraeder (mittlere Längenskalen).

Ein Ring ist dabei durch die kürzeste geschlossenen Verbindung aus Si-O-Brücken definiert, die im jeweils gleichen Siliziumatom starten und enden [111, 112, 113]. Dabei ist die Ringlänge  $n$  die Anzahl der Si-O Bindungen, die sich in einem solchen Ring befinden.

Beispielhaft gezeigt ist dies in Abbildung 2.12. Das mit dem zusätzlichen Ring markierte O-Si-O Element ist dabei der Ausgangspunkt der Betrachtung. Von dort aus existieren mehrere geschlossene Wege aus -Si-O- Elementen, um zum markierten Teil zurückzukehren. Der kürzeste geschlossene Weg über -Si-O- Elemente wird hier als Ring bezeichnet.

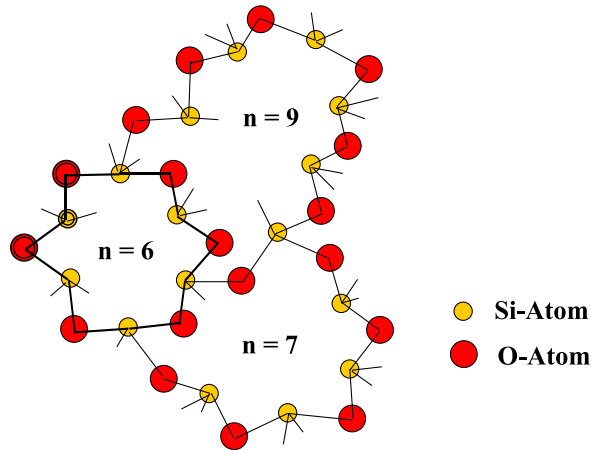


Abbildung 2.12.: Skizze zur Definition von Si–O Ringen der Länge  $n$ .

### 2.2.1.5. Die partiellen statischen Strukturfaktoren $S_{\alpha\beta}(q)$

Im Gegensatz zu den Paarkorrelationsfunktionen, die besonders gut geeignet sind, um Aussagen über Strukturen auf kleinen Längenskalen zu machen, ermöglichen die partiellen statischen Strukturfaktoren  $S_{\alpha\beta}(q)$ , die (bis auf eine Verschiebung um  $-1$ ) Fouriertransformierte von  $g_{\alpha\beta}(r)$  sind, strukturelle Informationen auf intermediären und großen Längenskalen besser zu erkennen. Diese Informationen sind zwar schon in den partiellen Paarkorrelationsfunktionen enthalten, werden aber erst durch die Fouriertransformation gut sichtbar.

Definiert sind die partiellen statischen Strukturfaktoren mittels  $g_{\alpha\beta}(r)$ , dem Wellenvektor  $\vec{q}$  und der Dichte  $\rho$  als

$$S_{\alpha\beta}(q) = 1 + \rho \int d\vec{r} e^{i\vec{k}\cdot\vec{r}} (g_{\alpha\beta}(r) - 1) \quad \text{mit} \quad |\vec{q}| = q; |\vec{r}| = r.$$

Unter Verwendung von Gleichung (2.42) erhält man

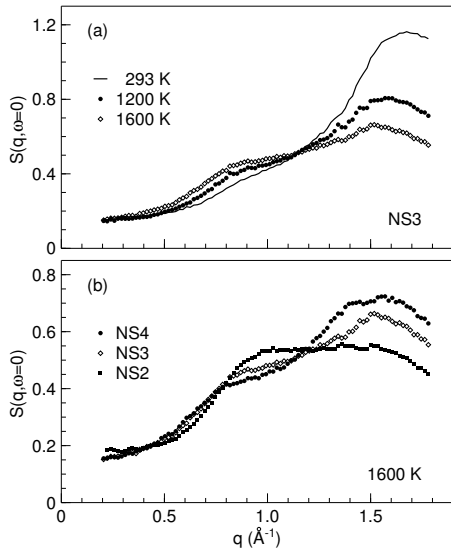
$$S_{\alpha\beta}(q) = \frac{f_{\alpha\beta}}{N} \left\langle \sum_{i=1}^{N_\alpha} \sum_{j=1}^{N_\beta} \exp(i\vec{q} \cdot \vec{r}_{ij}) \right\rangle \quad \alpha, \beta \in \{\text{Si, O, Li, K}\}, \quad (2.44)$$

mit  $N$ ,  $N_\alpha$  wie in Kapitel 2.2.1.1 und

$$f_{\alpha\beta} = \begin{cases} 1 & \alpha = \beta \\ \frac{1}{2} & \alpha \neq \beta \end{cases}.$$

Im Gegensatz zum Experiment<sup>13</sup> ist man bei Computersimulationen in der Lage, die partiellen Strukturfaktoren zu bestimmen. Die Bedeutung der Möglichkeiten, die man durch die Bestimmung partieller Strukturfaktoren erhält, wird klar, wenn man ein gelungenes Beispiel für das Ineinandergreifen von Experiment und Simulation betrachtet. In der Arbeit von Meyer et al.

<sup>13</sup>In Ausnahmefällen, z.B. bei  $\text{ZnCl}_2$  [116], ist es möglich, partielle Strukturfaktoren experimentell zu bestimmen.



**Abbildung 2.13.:** Elastischer Strukturfaktor aus Neutronenstreuungsmessungen [114, 115] für u.a. NS3 ( $(\text{Na}_2\text{O}) \cdot 3(\text{SiO}_2)$ ). Relativ unabhängig von der Zusammensetzung wird der Prepeak bei  $0.9 \text{ \AA}^{-1}$  mit steigender Temperatur ausgeprägter.

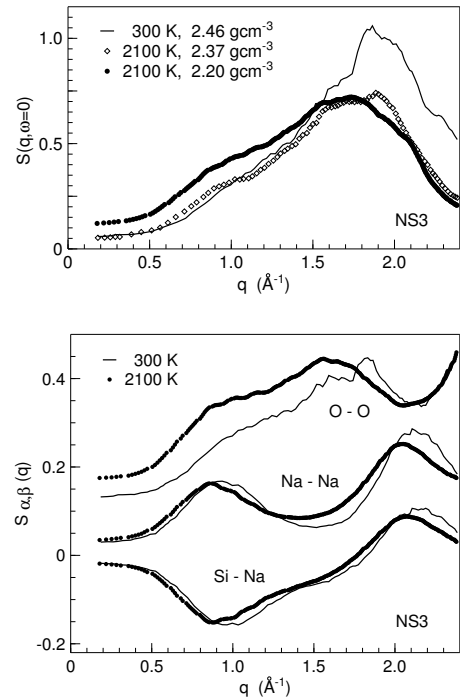
[114] werden einerseits elastische Strukturfaktoren (vgl. Kapitel 2.2.1.7) für  $(\text{Na}_2\text{O}) \cdot 3(\text{SiO}_2)$  aus dem Experiment, Abbildung 2.13, vorgestellt, andererseits werden diese mit den Vorhersagen aus der MD-Simulation, Abbildung 2.14, verglichen. Durch diese „Kreuzbefruchtung“ von Experiment und Simulation, in der die Simulation eine Vorhersage über die Existenz eines Prepeaks bei  $0.9 \text{ \AA}^{-1}$  machte, das Experiment diese bestätigte, war es dann wiederum möglich, durch aus der Simulation gewonnene mikroskopische Details (hier: partielle Strukturfaktoren) den experimentellen Befund zu deuten [114].

### 2.2.1.6. Der statische „Neutronenstreu“-Strukturfaktor $S_{\text{Neutron}}(q)$

Um Vergleichbarkeit mit Neutronenstreustrukturfaktoren aus dem Experiment zu erreichen, muß man die partiellen Strukturfaktoren aus Kapitel 2.2.1.5 mit den experimentellen Streulängen (siehe Tabelle 2.7) gewichten. Dann erhält man gemäß

$$S_{\text{Neutron}}(q) = \frac{1}{\sum_{\alpha} N_{\alpha} b_{\alpha}^2} \sum_{\alpha, \beta} b_{\alpha} b_{\beta} S_{\alpha\beta}(q) \quad \alpha, \beta \in \{\text{Si, O, Li, K}\} \quad (2.45)$$

**Abbildung 2.14.:** Oben: Elastischer Strukturfaktor der MD-Simulation für NS3. Unten: partielle Strukturfaktoren aus der MD-Simulation. Aus ihnen konnte der strukturelle Ursprung des Prepeaks ermittelt werden. Die Simulationen bei 300 K und 2100 K mit einer Dichte von  $\rho = 2.2 \text{ g/cm}^3$  sind mit den experimentellen Dichten durchgeführt worden.



Element	Streulänge [fm]
Si	4.1491
O	5.803
Li	-1.9
K	3.67

**Tabelle 2.7.:** Experimentelle Streulängen aus [117].

den statischen „Neutronenstreu“-Strukturfaktor.

### 2.2.1.7. Der elastische Strukturfaktor

Der elastische Strukturfaktor ist interessant, da diese Größe im Experiment durch inelastische Neutronenstreuung direkt gewonnen werden kann, ohne daß Annahmen zum Meßprozeß gemacht werden müssen. Eine gute Approximation des elastischen Strukturfaktors  $S(q, \omega = 0)$  ist durch [118, 119]

$$S(q, \omega = 0) \approx f(q) S_{\text{Neutron}}(q, t = 0) \quad (2.46)$$

gegeben, wobei  $f(q)$  der Debye–Waller–Faktor<sup>14</sup> ist.

## 2.2.2. Dynamische Größen

Hier sollen nun einige analytische Werkzeuge vorgestellt werden, die es möglich machen, die mikroskopische Dynamik zu untersuchen.

### 2.2.2.1. Mean Squared Displacement (MSD)

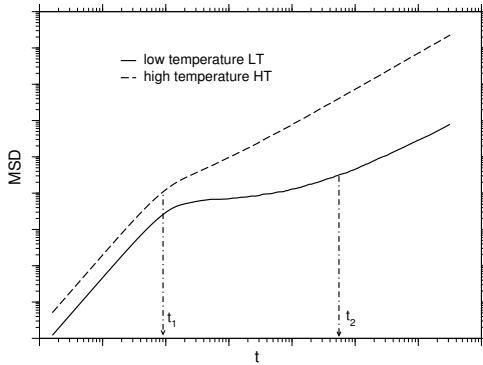
Eine der am einfachsten zu berechnenden Größen zur Untersuchung der Dynamik von Flüssigkeiten auf mikroskopischem Niveau ist das mittlere Verschiebungsquadrat (mean squared displacement oder kurz: MSD). Für eine Teilchensorte  $\alpha$  mit der Teilchenzahl  $N_\alpha$  lautet sie

$$\langle r_\alpha^2(t) \rangle = \frac{1}{N_\alpha} \sum_{i=1}^{N_\alpha} |\vec{r}_i(t) - \vec{r}_i(0)|^2 \quad \alpha \in \{\text{Si, O, Li, K}\}.$$

Ein typischer Verlauf des MSD für hohe und niedrige Temperaturen wird exemplarisch in Abbildung 2.15 gezeigt. Für hohe Temperaturen zeigt das MSD ein Zeitverhalten in zwei Regimes, nimmt die Temperatur ab, so bildet sich zwischen diesen ein Plateau aus, ein drittes Regime.

Das in beiden Fällen zeitlich zuerst auftretende Regime ist das **mikroskopische Regime**. Für kleine Zeiten ( $t < t_1$ ) bewegen sich die Teilchen wechselwirkungsfrei, also ballistisch. Daraus ergibt sich in diesem Regime die Bewegungsgleichung  $\vec{r}_i(t) \approx \vec{r}_i(0) + \vec{v}_i(0) \cdot t$  mit der thermischen Geschwindigkeit  $|\vec{v}_i| = \sqrt{(3k_B T)/m}$ , so daß das MSD proportional zu  $t^2$  ist.

<sup>14</sup>Der Debye–Waller–Faktor gibt bei Beugung und Streuung die Amplitudenschwächung aufgrund thermischer Schwankung an.



**Abbildung 2.15.:** Exemplarischer Verlauf von mittleren Verschiebungsquadraten in den in dieser Arbeit behandelten Systemen. Deutlich sichtbar: Im HT-Bereich ein Verhalten mit zwei Zeitregimes, im LT-Bereich sind drei Zeitregimes unterscheidbar.

Für niedrige Temperaturen beobachtet man zwischen dem mikroskopischen (ballistischen) Regime (s.o.) und dem diffusiven Bereich (s.u.) ein Plateau, ein Bereich in dem sich das MSD nicht signifikant verändert, daß  **$\beta$ -Relaxationsregime**. Dieses Verhalten kann man durch den sogenannten *Käfigeffekt* erklären. Anschaulich bedeutet es, daß ein Teilchen im „Käfig“ seiner Nachbarn gefangen ist (siehe Abbildung 2.16 a). Durchschnittlich braucht ein Teilchen die Zeitlänge des Plateaus, um sich aus diesem zu befreien (siehe Abbildung 2.16 b).

Schlußendlich, wenn die Dynamik nicht eingefroren ist, also nicht im  $\beta$ -Relaxationsregime verharret, geht das MSD über in den **diffusiven Bereich**. Auf langen Zeitskalen bewegen sich dort die Teilchen diffusiv und es gilt die Einstein-Relation

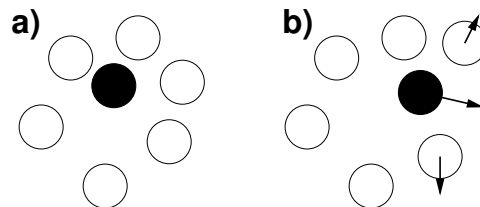
$$\langle r_\alpha^2(t) \rangle = 6D_\alpha t \quad (2.47)$$

mit der Diffusionskonstante  $D_\alpha$ . Diese Relation verbindet MSD und Diffusionskonstante, so daß man in der Lage ist, mithilfe des MSD Diffusionskonstanten zu bestimmen.

### 2.2.2.2. Die van Hove Korrelationsfunktionen $G_s(r)$ und $G_d(r)$

Zur Untersuchung der Art und Weise, wie Teilchen sich im System fortbewegen, z.B. durch Hopping oder kontinuierlich, und welche Positionen im System mit welchen Teilchen belegt werden, bieten sich die sogenannten *van-Hove-Korrelationsfunktionen* an. Sie sind für eine Teilchensorte  $\alpha$  (mit Teilchenzahl  $N_\alpha$ ) definiert über [6]

$$G^\alpha(r, t) = \frac{1}{N} \langle \rho(\vec{r}, t) \rho(\vec{0}, 0) \rangle = G_s^\alpha + G_d^\alpha \quad \alpha \in \text{Si, O, Li, K} \quad (2.48)$$



**Abbildung 2.16.:** *Käfigeffekt*: a) Teilchen gefangen im „Käfig“ seiner Nachbarn. b) Durch kollektive Bewegung entsteht eine Lücke, durch die das Teilchen „fliehen“ kann.

mit

$$G_s^\alpha(r, t) = \frac{1}{N_\alpha} \left\langle \sum_{i=1}^{N_\alpha} \delta(r - |\vec{r}_i(t) - \vec{r}_i(0)|) \right\rangle \quad \alpha \in \text{Si, O, Li, K} \quad (2.49)$$

und

$$G_d^{\alpha\beta}(r, t) = \mathcal{N}_{\alpha\beta} \left\langle \sum_{i=1}^{N_\alpha} \sum_{j=1, \text{ohne}(i)}^{N_\beta} \frac{1}{4\pi r^2} \delta(r - |\vec{r}_j(t) - \vec{r}_i(0)|) \right\rangle \quad \alpha, \beta \in \text{Si, O, Li, K} \quad (2.50)$$

wobei

$$\mathcal{N}_{\alpha\beta} = \begin{cases} \frac{N}{\rho N_\alpha (N_\alpha - 1)} & \alpha = \beta \\ \frac{N}{\rho N_\alpha N_\beta} & \alpha \neq \beta \end{cases}.$$

$4\pi r^2 G_s^\alpha(r, t)$  ist hierbei proportional zur Wahrscheinlichkeit, ein Teilchen, das zur Zeit 0 am Ort  $\vec{r}_i(0)$  war, zur Zeit  $t$  im Abstand  $r$  davon vorzufinden. Man sieht, daß hier auch die Information aus dem MSD enthalten ist. Tatsächlich stellt das MSD eine Untermenge der Informationen von  $G_s^\alpha(r, t)$  dar, genauer gesagt ist das MSD das zweite Moment von  $G_s^\alpha(r, t)$ .

$4\pi r^2 G_d^{\alpha\beta}(r, t)$  ist dagegen proportional zur Wahrscheinlichkeit, ein von  $i$  unterschiedliches Teilchen  $j$  zur Zeit  $t$  im Abstand  $r$  von dem Punkt anzutreffen, wo sich das Teilchen  $i$  zur Zeit 0 aufgehalten hatte. Auch hier ist die Information einer Größe, die schon definiert wurde, bereits enthalten, denn  $G_d^{\alpha\beta}(r, t = 0) \equiv g_{\alpha\beta}(r)$ .

### 2.2.2.3. (In)Kohärente intermediäre Streufunktionen $F_s(q, t)$ & $F(q, t)$

Durch Fouriertransformation der van-Hove-Korrelationsfunktionen erhält man die (in)kohärenten intermediären Streufunktion. Der Name rührt daher, daß diese Größe aus Neutronenstreuexperimenten gewonnen werden [119].

Zunächst soll hier die *inkohärente intermediäre Streufunktion* vorgestellt werden. Definiert ist sie durch

$$F_s^\alpha = \int G_s^\alpha(r, t) e^{-i\vec{q}\vec{r}} d\vec{r} = \frac{1}{N_\alpha} \sum_i^{N_\alpha} \langle \exp(-i\vec{q} \cdot [\vec{r}_i(t) - \vec{r}_i(0)]) \rangle. \quad (2.51)$$

Anschaulich ist sie ein Maß für die Lokalisierung einzelner Teilchen zur Zeit  $t$ , relativ zu dem Ort  $\vec{r}_i(0)$ . Fällt  $F_s^\alpha$  auf 0 ab, so gibt es keine Korrelationen mit den Ausgangspositionen der Teilchen mehr.

Die *kohärente intermediäre Streufunktion* hingegen ist eine Mehrteilchen-Korrelationsfunktion. Sie ist ein Maß für die Lokalisierung von Teilchen  $\beta$  in Plätzen (zur Zeit  $t$ ), in denen zur Zeit  $t = 0$  Teilchen der Sorte  $\alpha$  waren:

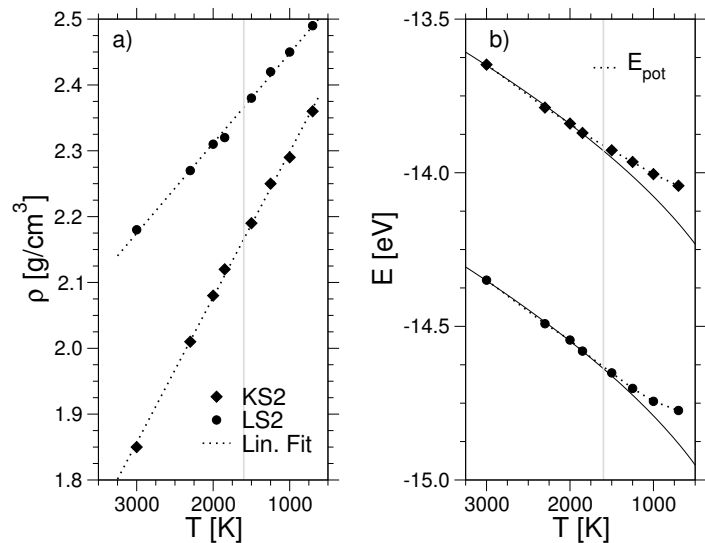
$$F^{\alpha\beta} = \frac{1}{N_\alpha N_\beta} \sum_i^{N_\alpha} \sum_j^{N_\beta} \langle \exp(-i\vec{q} \cdot [\vec{r}_j(t) - \vec{r}_i(0)]) \rangle. \quad (2.52)$$

Weiterhin kann man aus den (in)kohärenten Streufunktion die *Lamb-Mößbauer*<sup>15</sup> bzw. *Debye-Waller*-Faktoren gewinnen, indem man die Plateauhöhen der Streufunktionen abliest.

<sup>15</sup>Der Lamb-Mößbauer-Faktor ist der elastische Beitrag an der Streuung von Neutronen.

### 3. Eigenschaften von Alkalidisilikaten

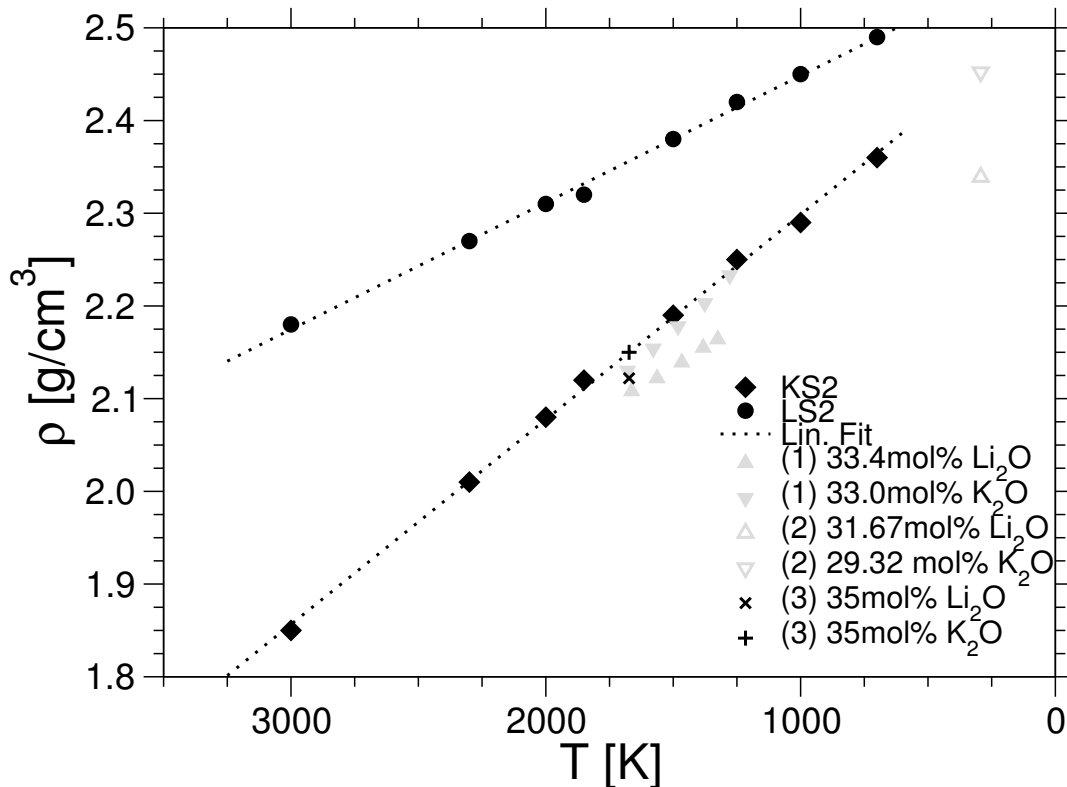
**Abbildung 3.1.:** a) Dichte als Funktion Temperatur. Die gestrichelten Linien sind ein linearer Fit der Daten. b) Totale und potentielle Energie in Abhängigkeit von der Temperatur. Die durchgezogenen Linien stellen die Vorhersage des Verlaufs der potentiellen Energie nach Rosenfeld and Tarazona [120] dar (Fitbereich 2000 K bis 3000 K). In grau wird  $T_g^{\text{sim}}$  angedeutet.



Im Rahmen der Untersuchungen zum MAE wurden auch zwei binäre Alkalisilikate simuliert, nämlich Lithiumdisilikat (LS2) und Kaliumdisilikat (KS2). Diese Auswahl ist darin begründet, daß einerseits das gewählte Potential<sup>1</sup> nur die Alkalispezies Lithium, Natrium und Kalium zu Verfügung stellt, andererseits ist aus Experimenten bekannt, daß der MAE mit Lithium und Kalium besonders ausgeprägt ist. Die Untersuchung der Eigenschaften dieser Systeme soll in diesem Kapitel im Vordergrund stehen.

Abgesehen vom Vergleich mit vorliegenden experimentellen Ergebnissen ist von speziellem Interesse, ob man für Alkalisilikate generische Eigenschaften feststellen kann, oder ob qualitative Unterschiede zwischen Alkalisilikaten mit verschiedenen Alkalikomponenten festzustellen sind. In diesem Zusammenhang ist auch der Vergleich mit Arbeiten zu Natriumisilikaten von Horbach [6] und Winkler [17] interessant, speziell da diese Simulationen mit einem anderen Potential, dem BKS-Potential (siehe Infokasten auf Seite 34), durchgeführt wurden. Gemeinsamkeiten der qualitativen Eigenschaften der Alkalisilikate mit den Ergebnissen aus letztgenannten Arbeiten würden Hinweise auf generische Eigenschaften geben.





**Abbildung 3.2.:** Vergleich der Dichteergebnisse aus der Simulation mit experimentellen Daten aus (1)=[121], (2)=[122] und (3)=[123].

### 3.1. Strukturelle Eigenschaften

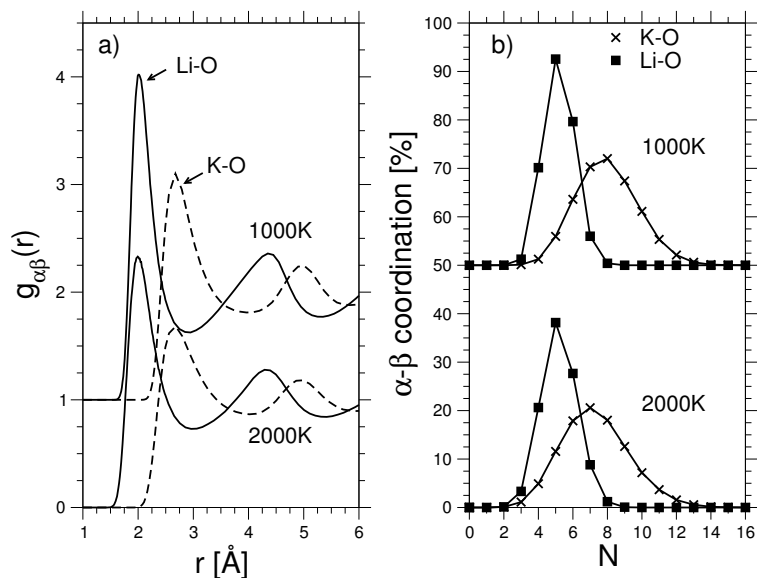
Betrachtet man Abbildung 3.1, so erkennt man sowohl für die Dichte als auch für die potentielle Energie einen über den untersuchten Temperaturbereich fast linearen Verlauf. Hier sollen die Dichtedaten aus der Simulation mit experimentellen Ergebnissen aus [121, 122, 123] (siehe Abbildung 3.2) verglichen werden. Betrachtet man zunächst die Daten für KS2, so stellt man eine fast perfekte Übereinstimmung der Dichtedaten der Simulation mit den experimentellen Ergebnissen fest. Trotz der Probleme, die das Kaliumpotential aufweist<sup>2</sup>, reproduziert es die Dichte von KS2 im Bereich  $T < 1700$  K sehr gut. Für LS2 hingegen wird nur das lineare Verhalten der Dichte qualitativ reproduziert. Die Dichte als Funktion der Temperatur liefert für fallende Temperaturen eine flachere Steigung als für KS2. Im Temperaturbereich  $1400 \text{ K} \leq T \leq 1700 \text{ K}$  ist die experimentelle Dichte von LS2 zudem in etwa  $0.25 \text{ g/cm}^3$  kleiner, als die in der Simulation bestimmte.

Beim Verlauf der potentiellen Energien in Abbildung 3.1.b) wurde die Vorhersage von Rosenfeld and Tarazona [120] mit den hier untersuchten binären Alkalidisilikaten verglichen. Rosenfeld and Tarazona leiten auf Basis des freien Energiefunktional in Verbindung mit harten Kugeln und thermodynamischer Störungstheorie analytisch ab, daß die potentielle Energie

<sup>1</sup>vgl. Kapitel 2.1.2.1.

<sup>2</sup>vgl. dazu Infokasten auf Seite 35.

**Abbildung 3.3.:** a) Partielle Paarkorrelationsfunktionen  $g_{\alpha\beta}(r)$  für Li – O und K – O. Daten für 1000 K um 1 nach oben verschoben. b) Histogramme der Koordinationszahlen. Daten für 1000 K um 50% nach oben verschoben. Gut zu erkennen sind die verschiedenen Längenskalen der Alkali–Sauerstoffabstände und die unterschiedliche Koordination mit Sauerstoff.



$U$  von Fluiden mit der Temperatur  $T$  wie  $U = a + b \cdot T^{3/5}$  skaliert. In Abbildung 3.1.b) wurde diese Funktion gegen die Daten im Temperaturbereich von 2000 K bis 3000 K der potentiellen Energie der Simulation gefittet (durchgezogene Linien). Man erkennt, daß zumindest für den hier untersuchten Temperaturbereich die Fitfunktionen nicht das beobachtete Verhalten reproduzieren. Für die hohen Temperaturen liegen die Fits auf den Simulationsdaten. Aber schon direkt unterhalb 1850 K zeigen die Simulationsdaten ein leicht konvexes Abknicken mit fallender Temperatur. Hier weichen die Energien von dem Verhalten, daß sie für hohe Temperaturen gezeigt haben, ab. Das ist aber auch der Temperaturbereich ( $T_g^{\text{sim}} \approx 1600$  K), in dem, wie später bei der Diskussion der Dynamik klar werden wird, die Systeme aus dem Gleichgewicht fallen. Dies könnte eine mögliche Erklärung dafür sein, daß die Vorhersage von Rosenfeld and Tarazona hier nicht mehr für niedrige Temperaturen stimmt.

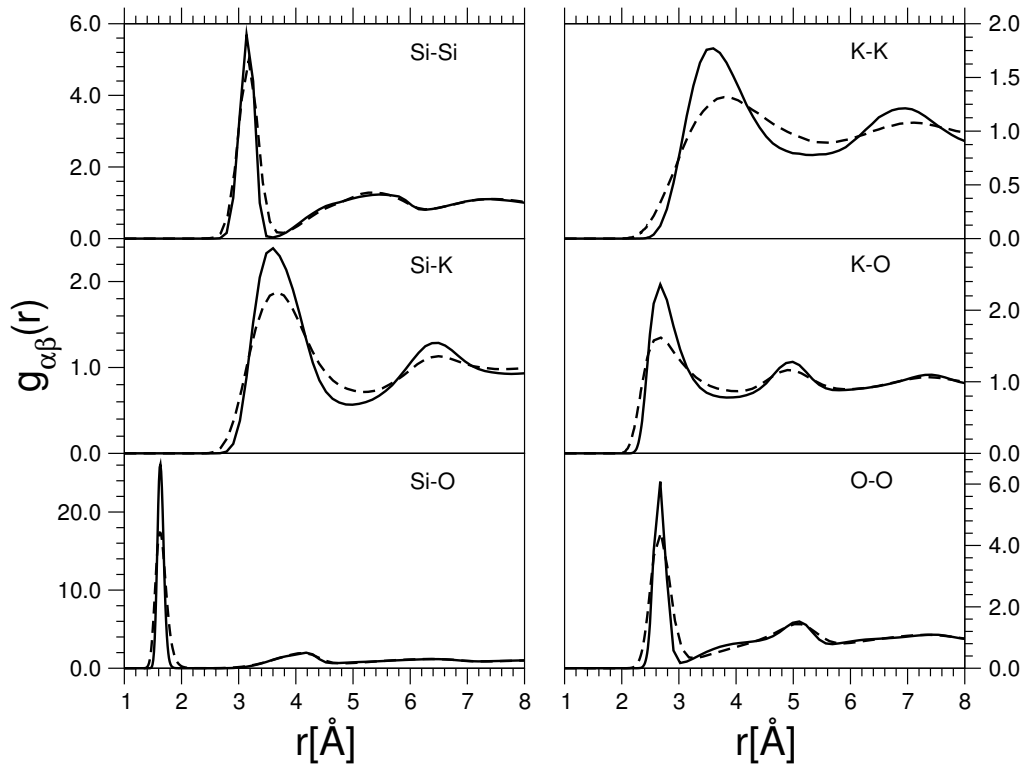
Zunächst soll hier die lokale Struktur mittels Paarkorrelationsfunktionen<sup>3</sup>  $g_{\alpha\beta}(r)$  untersucht werden. In Abbildung 3.3.a) ist  $g_{\alpha\beta}(r)$  für Alkali–Sauerstoff–Korrelationen bei den Temperaturen 1000 K und 2000 K dargestellt. Gut zu erkennen ist dabei, daß die Positionen der Peaks (und Minima) sich mit der Temperatur nur minimal verschieben, d.h., die hier sichtbaren Längenskalen ändern sich kaum mit der Temperatur, obwohl das System bei 2000 K noch komplett in allen Komponenten equilibriert ist, d.h. sich in der Schmelze befindet, und bei 1000 K nur die Alkaliionen sich im Gleichgewicht befinden, d.h. dort die Matrix aus dem Gleichgewicht gefallen ist (Glas). Allerdings unterscheiden sich Lithium– und Kalium–Korrelationen deutlich in den Längenskalen. Der Li–O–NN<sup>4</sup> Abstand<sup>5</sup> zum Beispiel beträgt 2.02 Å und der K–O Abstand 2.67 Å. Diese Unterschiede setzen sich in allen Paarkorrelationen, in denen Alkaliionen involviert sind, fort (siehe Abbildungen 3.4 und 3.5).

Gemeinsamkeiten in  $g_{\alpha\beta}(r)$  findet man speziell im Bereich der NN–Abstände der Matrixkomponenten, also für  $\alpha, \beta \in \{\text{Si}, \text{O}\}$ . Speziell die Paarkorrelationen  $g_{\text{SiO}}(r)$  sind für beide Systeme fast identisch und stimmen auch mit dem Wert überein, der in Simulationen zu rei-

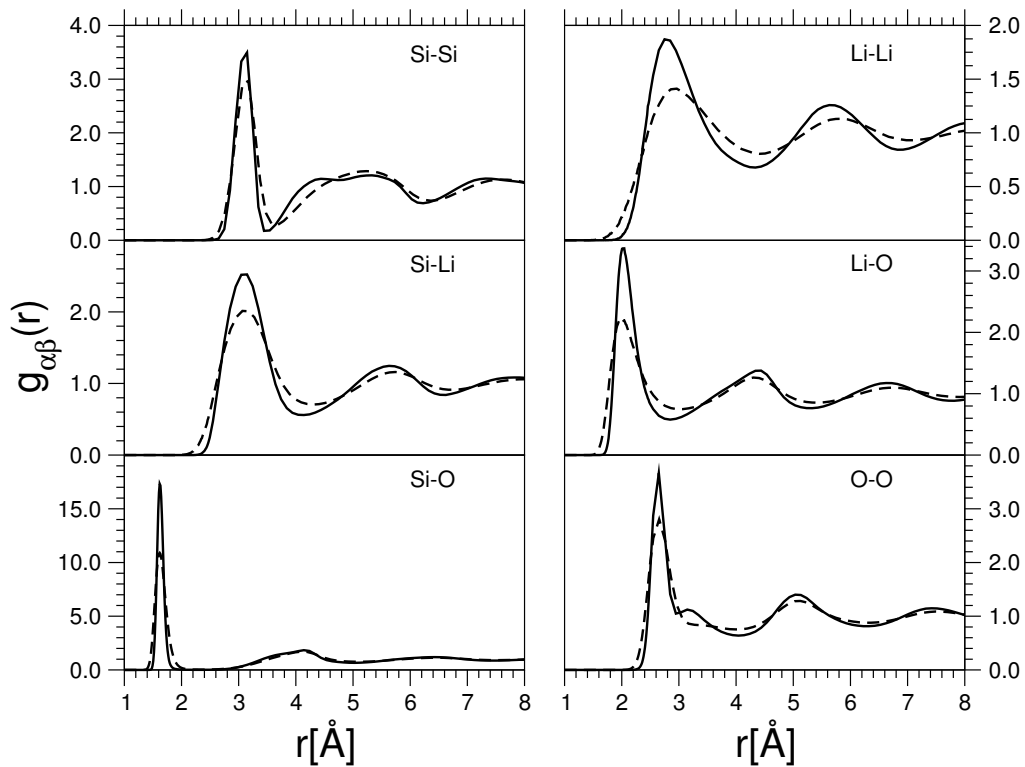
<sup>3</sup>vgl. Kapitel 2.2.1.1

<sup>4</sup>NN=nearest–neighbor bzw. nächster Nachbar.

<sup>5</sup>Eine Liste ausgewählter erster NN–Abstände findet sich in Tabelle 2.5.



**Abbildung 3.4.:** Übersicht der Paarverteilungsfunktionen von KS2 für 700 K (durchgezogene Linien) und 2300 K (gestrichelte Linien).



**Abbildung 3.5.:** Analoger Plot zu Abbildung 3.4 für LS2.

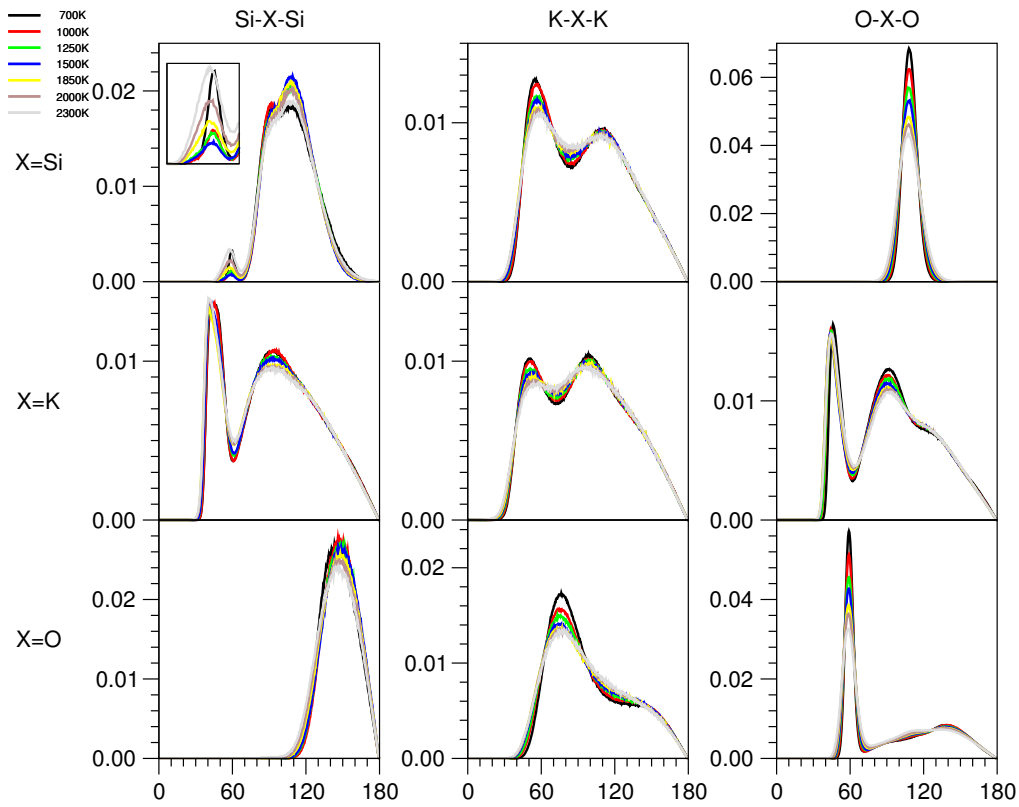
nem  $\text{SiO}_2$  und Natriumtrisilikat ermittelt wurde [6]. Das zeigt, daß das Vorhandensein von Alkaliionen die Bindungslängen in den  $\text{SiO}_4$ -Tetraedern nicht beeinflusst. Das dieses Verhalten aber falsch ist, zeigen sowohl Experimente, als auch Ab-Initio-Simulationen [73]. In ihnen beobachtet man eine Veränderung der Funktion  $g_{\text{SiO}}(r)$  bei Anwesenheit von Alkaliionen im System. Kommen sich Sauerstoff und Alkaliionen nahe, so kommt es zu Ladungsverschiebungen zwischen diesen Atomen. Alleine dadurch müßte es auch bei der Korrelation zwischen einem solchen Sauerstoff und Silizium zu einer Änderung der Bindungslänge kommen. Die eben geschilderte Beobachtung zeigt, wo sowohl das BKS-Potential als auch das hier gewählte Potential ihre Grenzen erreichen.

Auch bei Si-Si ist der erste NN-Peak nicht durch die Alkalinachbarn beeinflusst. Erst beim zweiten NN-Peak sieht man für LS2 bei 700 K eine Veränderung im Verlauf von  $g_{\text{SiSi}}(r)$ . Dort erkennt man, daß sich der zweite Peak in zwei Teile auftrennt, mit Maxima bei ca.  $r_1 = 4.45 \text{ \AA}$  und  $r_2 = 5.3 \text{ \AA}$ . Diese Aufteilung kann nicht durch eine Verschiebung von Teilchen entstanden sein, die bei 2300 K im Abstand des ersten Peaks lokalisiert waren, da der sich aus den Inter-Tetraederabständen ergibt. Auch passt die Aufteilung des zweiten NN-Peaks nicht zum  $g_{\text{SiSi}}(r)$  für reines  $\text{SiO}_2$  (vgl. [6]). Daher muß sich diese Doppelpeakaufteilung aus dem Einfluß der Alkaliionen ergeben.

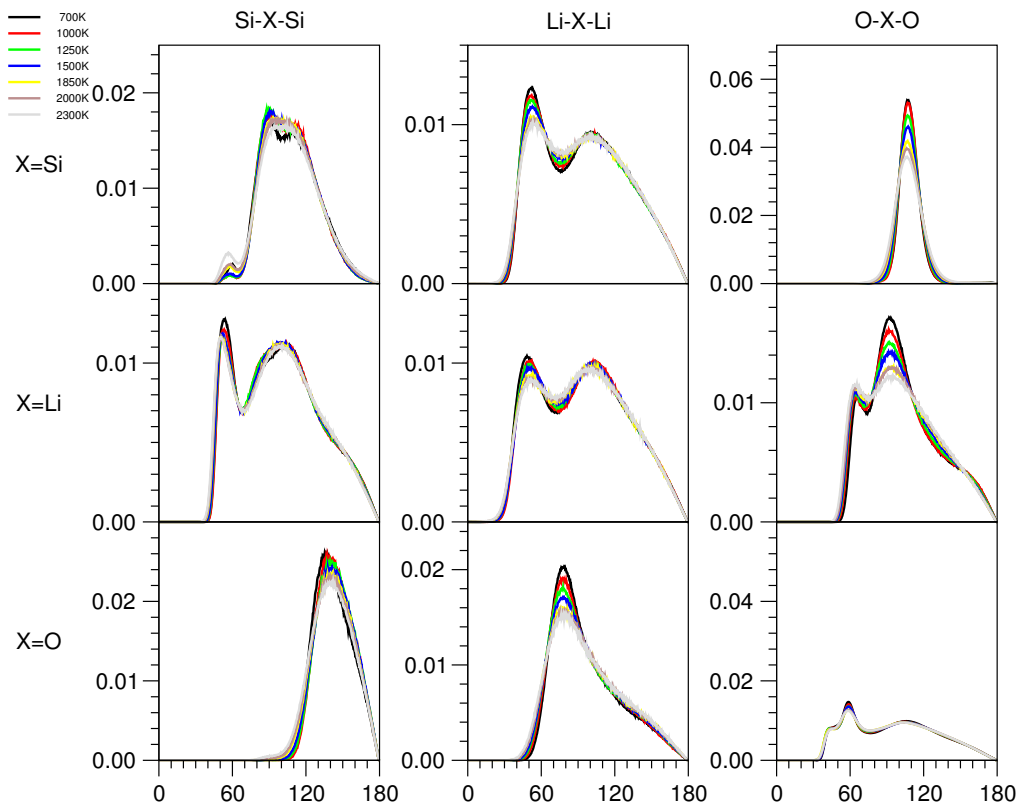
Erklären lässt sich diese Aufteilung mithilfe der mittleren Li-O-Abstände und der Bond-Bond-Winkelverteilung für O-Li-O (Abbildung 3.7). In der Winkelverteilung erkennt man für niedrige Temperaturen Peaks bei  $\phi_1 = 66^\circ$  und  $\phi_2 = 97^\circ$ , bei hohen Temperaturen verschmelzen diese Peaks fast zu einem Plateau. Der erste Nächste-Nachbar-Abstand für Li-O beträgt  $2.02 \text{ \AA}$ . Daraus kann man, wie in Abbildung 3.8 gezeigt, die Differenz der Längen der Verbindungsvektoren über Si-O-Li-O-Si für  $\phi_1$  und  $\phi_2$  bestimmen. Tatsächlich ergibt diese geometrische Betrachtung eine Längendifferenz von  $0.83 \text{ \AA}$ , was mit der Differenz  $r_2 - r_1 = 0.85 \text{ \AA}$  konsistent ist. Bei KS2 stechen keine gut sichtbaren Peaks hervor. Dabei ist es aber auch wichtig zu berücksichtigen, daß die Winkelverteilung für O-K-O über den untersuchten Temperaturbereich (2300 K bis 700 K) ihre Struktur wenig verändert, da die Peaks stärker getrennt sind als die Peaks bei  $\phi_1$  und  $\phi_2$  in LS2. Wenn dieser Anteil der Winkelverteilung Einfluß auf die Form von  $g_{\text{SiSi}}(r)$  hätte, so wäre dieser in allen untersuchten Temperaturen zu sehen.

Die nächste bemerkenswerte Beobachtung in den Paarverteilungen ist in der O-O-Paarkorrelation bei niedrigen Temperaturen, hier 700 K, zu finden. Bei LS2 (Abbildung 3.5) zeigt sich beim ersten Nächsten-Nachbar-Peak mehr als eine Schulter, eher ein „Nebenpeak“ um  $3.2 \text{ \AA}$ , bei KS2 (Abbildung 3.4) erkennt man eine zum zweiten Peak führende Schulter zwischen  $3.2 \text{ \AA}$  und  $4 \text{ \AA}$ . Wiederum weicht die Paarverteilung in diesen Punkten von der aus reinem  $\text{SiO}_2$  bekannten Form ab. Die Erklärung dafür findet man in den Winkelverteilungen. Zunächst für KS2: In der O-K-O Winkelverteilung (Abbildung 3.6) erkennt man für fallende Temperaturen ein Wachsen des Peaks im Bereich zwischen  $\phi_1 = 70^\circ$  und  $\phi_2 = 105^\circ$ . Das bedeutet allerdings, daß mithilfe der Argumentation, wie sie in Abbildung 3.8 skizziert ist, man ein Anwachsen der Paarkorrelationen im Bereich zwischen<sup>6</sup>  $r(\phi_1) = 2.67 \text{ \AA} \cdot 2 \cdot \sin(\phi_1/2) = 3.06 \text{ \AA}$  und  $r(\phi_2) = 4.24 \text{ \AA}$  hat. Damit erhält man die einleitende Schulter des zweiten NN-Peaks in  $g_{\text{OO}}(r)$  für KS2. Entsprechendes beobachtet man für LS2. Dort gilt  $r(\phi_1 = 79^\circ) = 2.02 \text{ \AA} \cdot 2 \cdot \sin(\phi_1/2) = 2.57 \text{ \AA}$  und  $r(\phi_2 = 110^\circ) = 3.31 \text{ \AA}$ . Das ist konsistent mit dem „Nebenpeak“

<sup>6</sup>Das erste Maximum von  $g_{\text{KO}}(r)$  in KS2 befindet sich bei  $2.67 \text{ \AA}$ , bei LS2 für  $g_{\text{LiO}}(r)$  bei  $2.02 \text{ \AA}$ .

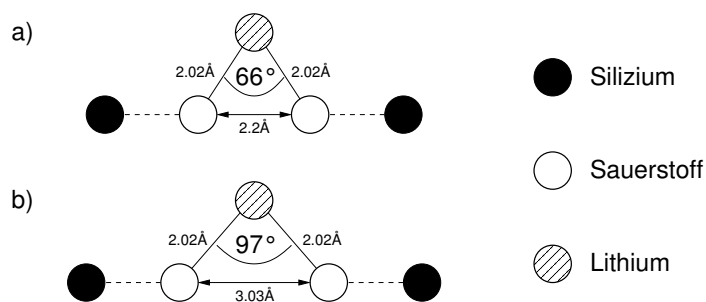


**Abbildung 3.6.:** Bond-Bond-Winkelverteilungen für KS2 bei Temperaturen von 700 K bis 2300 K. Erörterungen dazu ab Seite 55.



**Abbildung 3.7.:** Analoger Plot zu Abbildung 3.6, hier für LS2.

**Abbildung 3.8.:** Schema zur Erklärung der Aufspaltung des zweiten Nächste-Nachbar Peaks in Si–Si für niedrige Temperaturen in LS2 (vgl. Seite 53). Unterschied der Abstände zweier Si–Atome in einer Kette Si–O–Li–O–Si anhand verschiedener O–Li–O Bond–Bond Winkel, hier a)  $66^\circ$  und b)  $97^\circ$ .



des ersten NN–Maximums in  $g_{OO}(r)$ .

Das Vorhandensein sehr verschiedener Längen in der lokalen Struktur lässt auf verschiedene chemische Umgebungen um die Alkaliionen schließen. Bestätigt findet man das auch bei der Betrachtung der Koordinationszahlen<sup>7</sup>. In Abbildung 3.3.b) sind die Koordinationszahlverteilungen für die genannten Temperaturen aufgetragen. Erneut sieht man keine großen Änderungen in Abhängigkeit von der Temperatur. Andererseits sind die Koordinationen mit Sauerstoff für Lithium und Kalium sehr verschieden. Kalium ist im Mittel ca. 7.3– bis 8– (2000 K/1000 K) und Lithium ca. 5.2–fach mit Sauerstoff koordiniert.

Eine weitere untersuchte strukturelle Größe, mit der man intermediären Längenskalen untersucht, ist die Bond–Bond–Winkelverteilung (siehe Kapitel 2.2.1.3). Aus ihr kann man im Gegensatz zu isotropen Größen wie  $g_{\alpha\beta}(r)$  und den partiellen Strukturformfaktoren  $S_{\alpha\beta}(q)$  die lokale Richtungsabhängigkeit ablesen.

Hier soll nun zunächst die Winkelverteilung von KS2 besprochen werden, um sie dann mit der von LS2 zu vergleichen.

Bei KS2 (Abbildung 3.6) fallen einige Besonderheiten ins Auge. Bei allen Winkelverteilungen  $\alpha$ -K- $\alpha$  ( $\alpha \in \{\text{Si}, \text{K}, \text{O}\}$ ) erkennt man eine Doppelpeakstruktur mit dem ersten Peak bei  $45$ – $50^\circ$ , dem zweiten bei ca.  $95^\circ$  und einer (schwachen) Schulter bei  $130$ – $135^\circ$ . Das bedeutet, daß Kalium gleichnamige NN–Paare in diesen Winkeln bevorzugt.

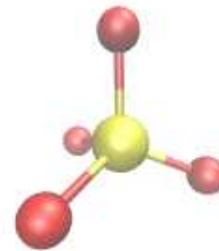
In der Verteilung Si–Si–Si sieht man bei der Betrachtung des ersten Peaks bei ca.  $58^\circ$  (siehe Inset), daß ein breiter Peak (grau) bei hoher Temperatur zunächst abnimmt, um dann ab ca. 1500 K (blau) wieder zu wachsen, diesmal aber schärfer definiert. Ein ähnliches Verhalten beobachtet man im Bereich zwischen ca.  $92^\circ$  und  $108^\circ$ . Es handelt sich dort um zwei sich stark überlappende Peaks. Bei 2300 K dominiert noch der Peak bei  $108^\circ$ . Die Höhe des Peaks steigt bei fallender Temperatur bis 1500 K an, um dann zugunsten des Peaks bei  $92^\circ$  abzufallen. Bei 700 K (schwarz) erhält man im Bereich zwischen den Peaks ein Plateau, d.h., die Höhen sind näherungsweise gleich. Es findet also eine Umverteilung in der Winkelverteilung der Si–Si–Si–Bonds statt, die sich nicht einfach linear mit der Temperatur erklären lässt. Auch Si–O–Si zeigt ein Verhalten, das nicht proportional mit der Temperatur verläuft. Die Position des Maximums der Verteilung verläuft von  $146.9^\circ$  bei 2300 K bis zu  $149.4^\circ$  bei 1500 K und dann wieder zurück zu  $145.6^\circ$  bei 700 K.

<sup>7</sup>Definition in Kapitel 2.2.1.2.

	Peakort (FWHM) [°]		
	Experiment	Simulation	
	[124]	[72]	diese Arbeit
Si–O–Si	144 (38)	142.151 (35.7)	145.6 (37.8)
O–Si–O	109.5	108.3 (12.8)	108.3 (13.32)

**Tabelle 3.1.:** Vergleich von Bond–Bond–Winkeln zwischen Experiment und Simulation von amorphem SiO<sub>2</sub>. Soweit vorhanden, werden die Halbwertsbreiten (FWHM) in Klammern angegeben. Die Simulation von Vollmayr [72] verwendete im Gegensatz zu dieser Arbeit das BKS–Potential [101]. Die Daten aus dieser Arbeit beziehen sich auf das KS2 System bei 700 K.

**Abbildung 3.9.:** SiO<sub>4</sub>–Tetraeder: das Si–Atom ist gelb gezeichnet, Sauerstoff rot. Ein großer Teil der Korrelationen des ersten NN–Peaks in  $g_{OO}(r)$  sowie der Bond–Bond–Winkelverteilung von O–O–O um 60° stammt von den Sauerstoffen, die tetraedisch in die Matrix eingebunden sind. Jeweils drei O–Atome um ein Si–Atom bilden idealerweise ein gleichseitiges Dreieck, daher auch die starke Winkelkorrelation um 60°.



Am Beispiel des Peaks um ca. 58° wird allerdings klar, weshalb in dieser Arbeit keine weiterführenden Aussagen über die gerade beschriebenen Phänomene gemacht werden können. Dieser Peak kann direkt mit Ringen der Länge  $n=3$  korreliert werden (vgl. Kapitel 2.2.1.4 und [73]). Von diesen gibt es aber in den untersuchten Systemen nur ca. 20 bis 60. Für jede Temperatur wurden nur ein bis zwei Simulationsläufe durchgeführt. Das heißt aber, daß viel zu wenig Statistik vorhanden ist, um zu sagen, ob es sich hierbei wirklich um einen Effekt handelt oder nur um statistisches „Rauschen“. Der Rechenaufwand, um dies zu untersuchen, ist zur Zeit der Erstellung dieser Arbeit zu hoch.

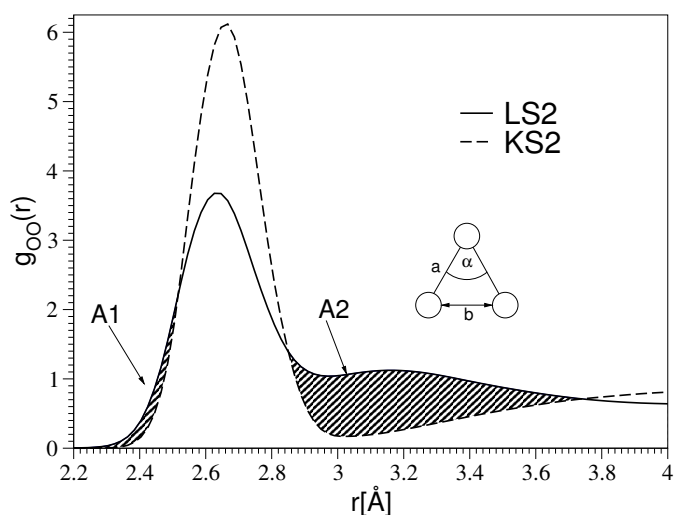
Wie nicht anders zu erwarten, da die Alkaliionen zwar die tetraedrische Struktur des Si–O–Netzwerks aufbrechen, nicht jedoch die einzelnen Tetraeder an sich beeinflussen, liegt das Maximum der O–Si–O–Winkel scharf gepeakt bei zunächst 107.7° (2300 K), verschiebt sich dann bei fallender Temperatur in Richtung des idealen Tetraederwinkels<sup>8</sup> und beträgt 108.3° bei 700 K. Die Werte der O–Si–O– und Si–O–Si–Winkel sind in Tabelle 3.1 aufgetragen und werden u.a. mit experimentellen Werten verglichen. Man erkennt die gute Übereinstimmung der Werte, speziell auch mit dem Experiment.

Weiterhin fällt hier bei O–O–O ein stark ausgeprägter Peak bei ca. 59° auf. Dieser läßt sich direkt auf die tetraedrische Grundstruktur der SiO<sub>2</sub>–Matrix zurückführen (vgl. dazu Abbildung 3.9).

LS2 zeigt im Vergleich generell ähnliche Eigenschaften. Peakhöhen und Maxima unterscheiden sich leicht von KS2 (siehe Abbildung 3.7). In den Li– $\beta$ –Li ( $\beta \in \{\text{Si}, \text{Li}, \text{O}\}$ ) erkennt man nun etwas stärkere Schultern bei großen Winkeln, die um 140° betragen.

<sup>8</sup>Der ideale Tetraederwinkel beträgt 109°28'.

**Abbildung 3.10.:** Überlagerung von  $g_{OO}(r)$  von KS2 und LS2 bei 700 K. Schraffiert ist die Differenzfläche, in der  $g_{OO}(r)$  in LS2 größer ist als  $g_{OO}(r)$  in KS2. Damit ist es möglich, die Schulter bei kleinen Winkeln in der O–O–O Winkelverteilung von LS2 zu erklären (vgl. Seite 57).



a [Å]	b [Å]	$\alpha$ [°]
2.85	2.26	46.72
2.85	2.5	52.03
3.73	2.26	35.27
3.73	2.5	39.16

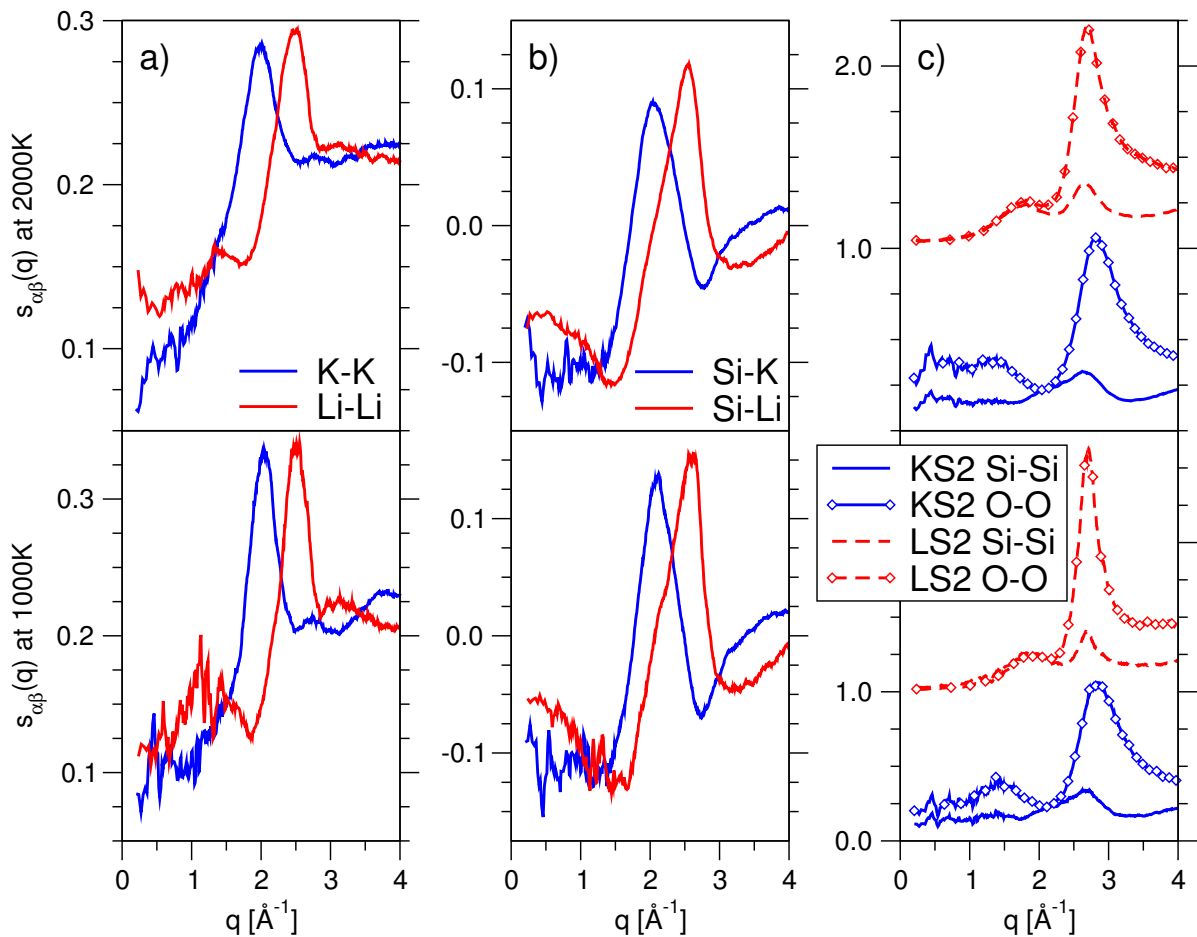
**Tabelle 3.2.:** Berechnung der Winkel  $\alpha$  aus den Längen a und b mittels eines gleichschenkeligen Dreiecks (a Länge der Schenkel) wie in Abbildung 3.10 definiert.

Dramatische Unterschiede sieht man jedoch bei den O– $\beta$ –O–Winkeln. O–Si–O zeigt eine deutlich breitere und weniger stark gepeakte Verteilung als in KS2. Bei 700 K beträgt die Halbwertsbreite  $16.5^\circ$ , das Maximum ist nur ca.  $4/5$  so hoch, wie in KS2 bei gleicher Temperatur. In KS2 zeigt in der Verteilung für O–K–O der erste Peak bei ca.  $45^\circ$  ein höheres Maximum als der darauffolgende Peak bei ca.  $92^\circ$ . Zudem sind diese Peaks durch ein ausgeprägtes Maximum getrennt. In LS2 zeigt die analoge Winkelverteilung von O–Li–O ein anderes Verhalten. Der erste Peak verschiebt sich zu ca.  $66^\circ$ , dadurch entsteht ein starker Überlapp zugunsten des zweiten Maximums bei ebenfalls ca.  $92^\circ$ .

Das Verhalten der O–O–O–Winkel in LS2 ist sehr von dem in KS2. Der Peak bei  $59^\circ$  ist in LS2 sehr schwach ausgeprägt, weniger als  $1/3$  so hoch wie in KS2. Dafür zeigt sich in LS2 eine ausgeprägte Schulter um  $43^\circ$ , die in KS2 nicht sichtbar ist. Hierbei kann wiederum eine geometrische Überlegung hilfreich sein. Da diese Schulter in KS2 nicht sichtbar ist, ist es sinnvoll, die Unterschiede der O–O–Paarkorrelationen in KS2 und LS2 hervorzuheben (bei 700 K). Dies wurde in Abbildung 3.10 getan. Die schraffierten Flächen beschreiben dabei die Differenzfläche zwischen den beiden Paarkorrelationen links und rechts vom ersten Peak der O–O–Paarkorrelation in KS2. Nimmt man jetzt die  $x$ -Koordinaten vom linken sowie rechten Ende der Flächen A1 und A2 und rechnet (Tabelle 3.2) die resultierenden Winkel  $\alpha$  aus, indem man gleichschenklige Dreiecke annimmt, wie in Abbildung 3.10 skizziert<sup>9</sup>, so erhält man einen Winkelbereich von ca.  $39^\circ$  bis  $52^\circ$ . Dies entspricht dem Winkelbereich der Schulter, die in der Bond–Bond–Winkelverteilung O–O–O von LS2 auftritt. Daß diese Schulter nicht in KS2 zu beobachten ist, ist auch klar. Hier verschiebt sich alles zu größeren Längenskalen.

<sup>9</sup>Hierbei wurde nur  $b < a$  untersucht, da die Ursache für kleine Winkel  $\alpha$  gesucht wurde.





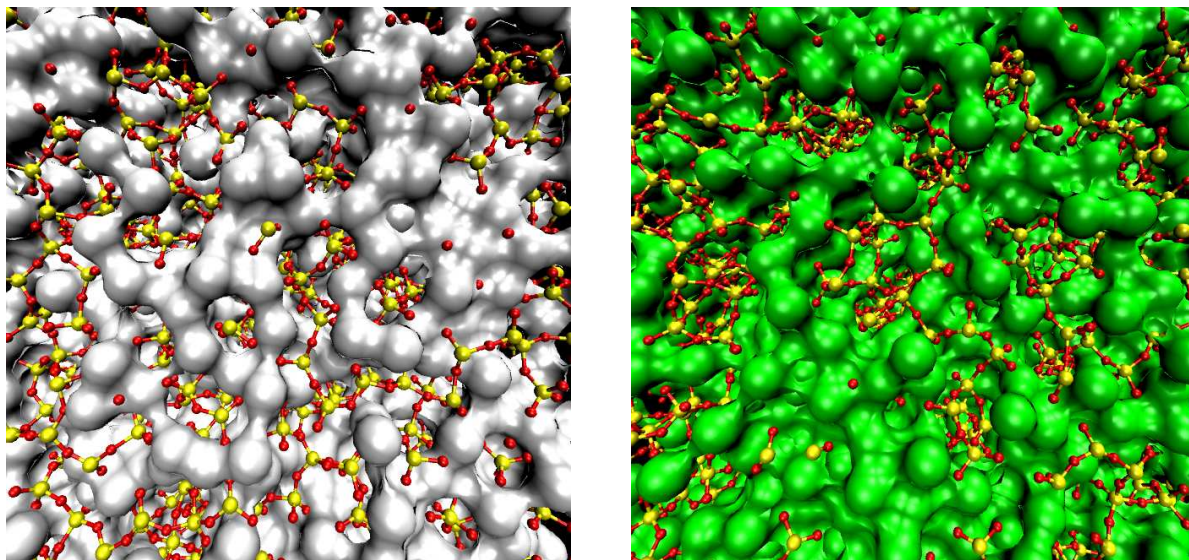
**Abbildung 3.11.:** Auswahl partieller Strukturfaktoren bei 1000 K und 2000 K.

Das bedeutet in diesem Fall, daß der „Nebenpeak“, der bei  $g_{OO}(r)$  in LS2 am ersten Peak zu erkennen ist, sich wie vorher erwähnt zu größeren Radien verschiebt. Das heißt aber auch, daß dieser Peak in der Winkelverteilung bei größeren Winkeln liegt, wo er mit bereits anderweitig vorhandenen Peaks verschmilzt.

Zusammenfassend kann man hier folgern, daß die Winkelverteilungen zwar für die beiden binären Alkalisilikate durchaus viele Analogien aufweisen, sich jedoch in einigen wichtigen Details, speziell in ihrer Nachbarschaft zu Sauerstoff, stark unterscheiden. Dies bekräftigt die Beobachtungen, die vorher schon bei den Paarkorrelationen und den Koordinationszahlen gemacht wurden: Die Sauerstoffumgebung um Alkaliionen ist zwischen verschiedenen Spezies von Alkaliatomen auch deutlich verschieden.

Die partielle Paarkorrelationsfunktion  $g_{\alpha\beta}(r)$  bietet allerdings nur einen optisch gut sichtbaren Eindruck für kurze Abstände. Auf intermediären und großen Längenskalen ist die Betrachtung der partiellen Strukturfaktoren  $S_{\alpha\beta}(q)$  aussagekräftiger.

Neben den Peaks mit der größten Amplitude in Abbildung 3.11, die um den  $q$ -Vektor der NN-Abstände liegen, erkennt man bei kleineren  $q$  (hier:  $\lesssim 2 \text{ \AA}^{-1}$ ) noch Peaks, die sogenannten Prepeaks. Besonders gut sind diese in Abbildung 3.11.a) zu erkennen. Für LS2 findet sich der



**Abbildung 3.12.:** Schnappschüsse von LS2 (links) und KS2 (rechts) bei 1000 K. Beschreibung der Darstellung im Text auf Seite 59. Generiert wurden alle Schnappschüsse in dieser Arbeit mit VMD [125].

Prepeak bei  $q_1 = 1.2 \text{ \AA}^{-1}$  und für KS2 bei  $q_{2a} = 0.7 \text{ \AA}^{-1}$  und  $q_{2b} = 0.5 \text{ \AA}^{-1}$ . Diese Längen korrespondieren mit den Abständen übernächster Alkali–Alkali– bzw. Si–Alkali–Nachbarn, wie sie analog für diverse Natriumsilikate bereits gefunden wurden [15, 14, 17]. Bei den partiellen Paarkorrelationen erkennt man bei KS2 oftmals für  $q \rightarrow 0$  eine große Amplitude. Dies allein ist schon ein Hinweis, daß KS2 relativ stark kompressibel ist und zum Entmischen neigt. Dazu kommt aber noch die Doppelpeakstruktur bei kleinen  $q$ , die sich auch in dem entmischenden System  $(\text{Na}_2\text{O})(2\text{SiO}_2)$  zeigt [59].

Eine Bestätigung, daß dies kein Artefakt der Simulation ist, erhält man aus elastischen Neutronenstreuexperimenten, bei denen bei entsprechendem Wellenvektor eine ausgeprägte Schulter sichtbar ist (vgl. Abbildung 2.7).

Das Vorhandensein dieser Prepeaks ist darin begründet, daß die Alkaliionen sich nicht vollkommen homogen im System anordnen. Klarheit über die Anordnung der Ionen erhält man durch Betrachtung von „Schnappschüssen“ der Konfigurationen der Simulation. Solche sind in Abbildung 3.12 zu sehen. Die Darstellung zeigt Silizium in gelb, Sauerstoff in rot, Lithium in weiß und Kalium in grün. Für Silizium und Sauerstoff wurde eine Darstellung<sup>10</sup> gewählt, in der die Atome als Kugeln und Bindungen als Zylinder gezeichnet sind. Gut zu sehen ist die tetraedrische Anordnung dieser Atomsorten. Dieser Teil des Systems wird später auch als „Matrix“ bezeichnet.

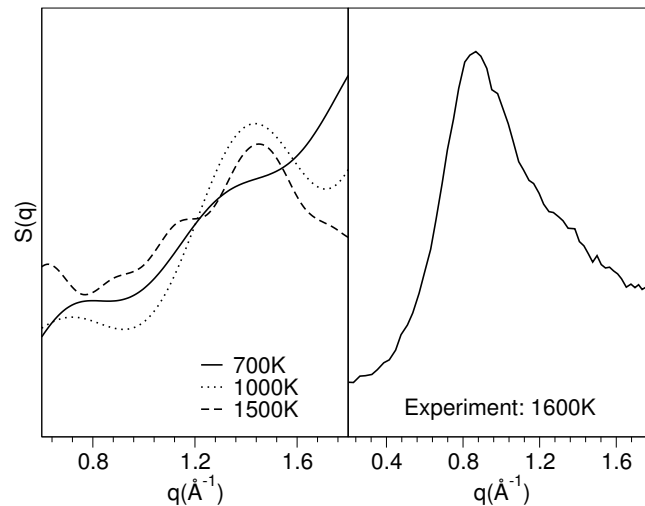
Für die Alkaliionen wurde hingegen eine Darstellung<sup>11</sup> verwendet, bei der im wesentlichen Kugeln um die Atome gelegt werden, die bei Annäherung aneinander „verschmelzen“. Der große Vorteil dieser Darstellung ist, daß man die Struktur, die die Alkaliionen ausbilden, so besser erkennen kann, nämlich die Ausbildung eines perkolationsartigen Kanalnetzwerks.

Nun kann man auch die in Abbildung 3.11 vorhandenen Prepeaks mit der optischen Beob-

<sup>10</sup>In VMD wird diese Darstellung CPK genannt.

<sup>11</sup>In VMD wird diese Darstellung *surf* genannt. Der gewählte Radius beträgt  $1.2 \text{ \AA}$ .

**Abbildung 3.13.:** Elastische Struktur­faktoren aus Simulationen zu KS2. Links die Ergebnisse aus dieser Arbeit, rechts die experimentellen Resultate von Kargl et al. [126]. Deutlich erkennbar ist der unterschiedliche Verlauf der elastischen Struktur­faktoren zwischen Experiment und Simulation. Hier zeigt sich, daß das Simulationspotential für Kalium das reale Verhalten schlecht beschreibt.



achtung der Schnappschüsse in Einklang bringen. Der Wellenvektor  $q_1$  entspricht der Länge  $5.24 \text{ \AA}$  und  $q_{2a/b}$  entsprechen ca.  $9\text{--}12 \text{ \AA}$ . Im linken Bild von Abbildung 3.12 erkennt man beim Lithiumdisilikat eine feinere Struktur, d.h. kleinere Längenskalen, zwischen den Alkali­ionenkanälen als im rechten Bild. Dort, im Kaliumdisilikat, erkennt man größere „Lochstruk­turen“<sup>12</sup>, was im Einklang ist mit den Unterschieden in den eben genannten Wellenvektoren, bei denen die Prepeak–Maxima auftreten.

Ein direkter Vergleich mit dem Experiment ist mittels eines Vergleichs mit den elastischen Neutronenstreu–Struktur­faktoren möglich. In Kapitel 2.2.1.7 wird beschrieben, wie man diese aus den Simulationsdaten erhält. In den Abbildungen 2.7 und 2.8 wurden bereits die Daten von Experiment und Simulation in LS2 gegenübergestellt. Dort erkennt man bei der Simulation einen Prepeak bei etwas höheren  $q$ –Werten (ca.  $1 \text{ \AA}^{-1}$ ), als im Experiment (ca.  $0.9 \text{ \AA}^{-1}$ ). Qualitativ kann man hier von einer recht guten Übereinstimmung mit dem Experiment reden.

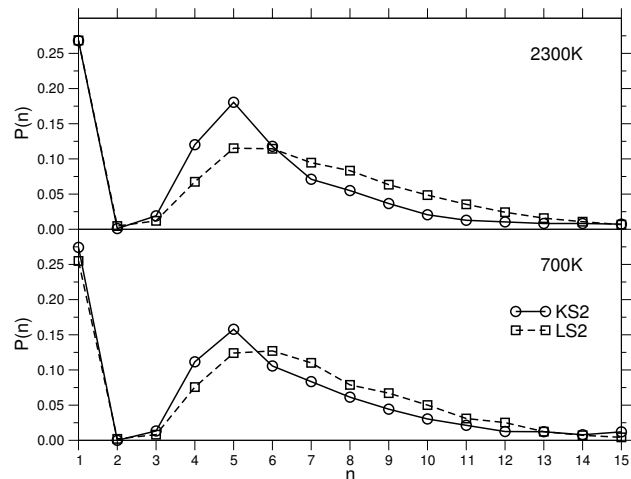
Anders jedoch das Bild bei KS2 (siehe 3.13). Hier unterscheiden sich experimentelle Ergebnisse und Simulationsresultate völlig. Dies ist offensichtlich eine Schwäche des verwendeten Potentials auf intermediären Längenskalen, denn man kann in Tabelle 2.5 sehen, daß die lokale Struktur, also die ersten Maxima von  $g_{\alpha\beta}(r)$ , sehr gut reproduziert werden.

Eine weitere Möglichkeit, intermediäre Längenskalen zu untersuchen, sind die Ringlängen­verteilungen<sup>13</sup> (Abbildung 3.14). Im Vergleich zu früheren Untersuchungen zu Ringlängen in Simulationen zu reinem  $\text{SiO}_2$  [73, 100, 111, 112] erkennt man sowohl bei KS2 als auch LS2 eine deutliche Verlagerung der Ringlängen zu höheren  $n$ . Zwar beobachtet man die Maxima der Verteilung immer noch im Bereich zwischen  $n=5$  (KS2) und  $n=6$  (LS2), aber die Anzahl der Ringe mit Längen  $n>7$  nimmt stark zu. Dieses Verhalten wurde bereits in Computersimulationen zu Natriumsilikaten (unter anderem Natriumdisilikat (NS2)) [17] beobachtet. Der Grund ist in dem Aufbrechen der  $\text{SiO}_2$ –Matrix zu suchen. Die Alkaliatome brechen die Netzwerkstruktur auf (Netzwerk­wandler). Dadurch entstehen auch die Ringe der Länge  $n=1$ , welche sogenannte NBOs (Non–Bridging–Oxygens) darstellen, also Sauerstoffatome, die zwar auf der einen Seite fest mit Silizium gebunden sind, auf der anderen Seite aber mit Alkali­ionen koordinieren. Dadurch werden aber viele mögliche „Wege“ innerhalb der Matrix beendet,

<sup>12</sup>Die Analogie zu einem Schweizer Käse ist durchaus hilfreich bei der Vorstellung.

<sup>13</sup>Definition der Ringlängenverteilung in Kapitel 2.2.1.4.

**Abbildung 3.14.:** Histogramme der Ringlängenverteilungen von KS2 und LS2 für 2300 K (oben) und 700 K (unten). Zwei Dinge sind klar zu erkennen: Kaum Änderung der Verteilung mit der Temperatur und eine klare Bevorzugung von 5-er Ringen in KS2, wohingegen LS2 eher 6-er Ringe enthält.



die im reinen  $\text{SiO}_2$  ein Schließen von kurzen Ringen erlaubt. Daher müssen viel längere Wege gesucht werden, um einen Ring zu schließen. Das erklärt auch, warum in LS2, welches, wie gerade gezeigt, kleinere intermediäre Längenskalen aufweist als KS2, höhere Ringlängen bevorzugt werden als in KS2: Die kleineren Längenskalen bedeuten ein engmaschigeres Kanalnetzwerk. Das wiederum impliziert kleinere  $\text{SiO}_2$ -Matrix-Strukturen zwischen den Kanälen, was nach der vorherigen Argumentation auch längere Wege zum Schließen von Ringen bedeutet.

Zusammenfassend stellt man fest, daß die binären Alkalidisilikate im statischen Verhalten recht ähnlich sind. Die Alkaliionen bilden hier perkolierende Netzwerke in der  $\text{SiO}_2$ -Matrix aus. Der Hauptunterschied, je nach Alkaliionensorte, ist jedoch das Ausbilden von strukturellen Korrelationen auf völlig verschiedenen Längenskalen, und zwar sowohl auf kleinen- (NN-Abstände) wie auf intermediären Skalen (Prepeaks). Zudem stellen sich um jede Alkaliionensorte dann auch andere chemische Umgebungen (O-Koordinationen) ein.

Die Frage, die sich nun stellt, ist wie und ob diese Unterschiede sich auf die Dynamik der Systeme auswirkt. Gibt es dort auch Unterschiede, oder stellen wir auch generische Eigenschaften fest, die sich nur in quantitativen Größen unterscheiden?

## 3.2. Dynamische Eigenschaften

Zunächst werden hier die mittleren Verschiebungswerte untersucht. In Abbildung 3.15 sind diese für KS2 und LS2 und Temperaturen von 700 K bis 3000 K doppelt logarithmisch aufgetragen. Man erkennt sofort, daß bei den tiefen Temperaturen sich zwischen ballistischem Regime und diffusivem Bereich für alle Komponenten ein deutliches Plateau ausbildet<sup>14</sup>. Zudem erkennt man, daß die Alkaliionen sich in gleicher Zeit weiter von ihrem Ursprung entfernen, als dies für die Matrixkomponenten Si und O der Fall ist. Speziell die Matrixkomponenten erreichen bei den tiefsten Temperaturen innerhalb der Simulationszeit nicht das diffusive Regime, sind also nicht im Gleichgewicht. Die Alkaliionen jedoch erreichen in allen simulierten Systemen das diffusive Regime.

Bei steigender Temperatur wird die Zeitdauer, in der sich die Teilchen im Plateaubereich befinden, kürzer. Das bedeutet, sie befreien sich schneller aus dem Käfig ihrer Nachbarn. Das führt bei weiter steigenden Temperaturen dazu, daß die Teilchen direkt aus dem ballistischen Regime in den diffusiven Bereich übergehen, ohne durch den Käfigeffekt aufgehalten zu werden.

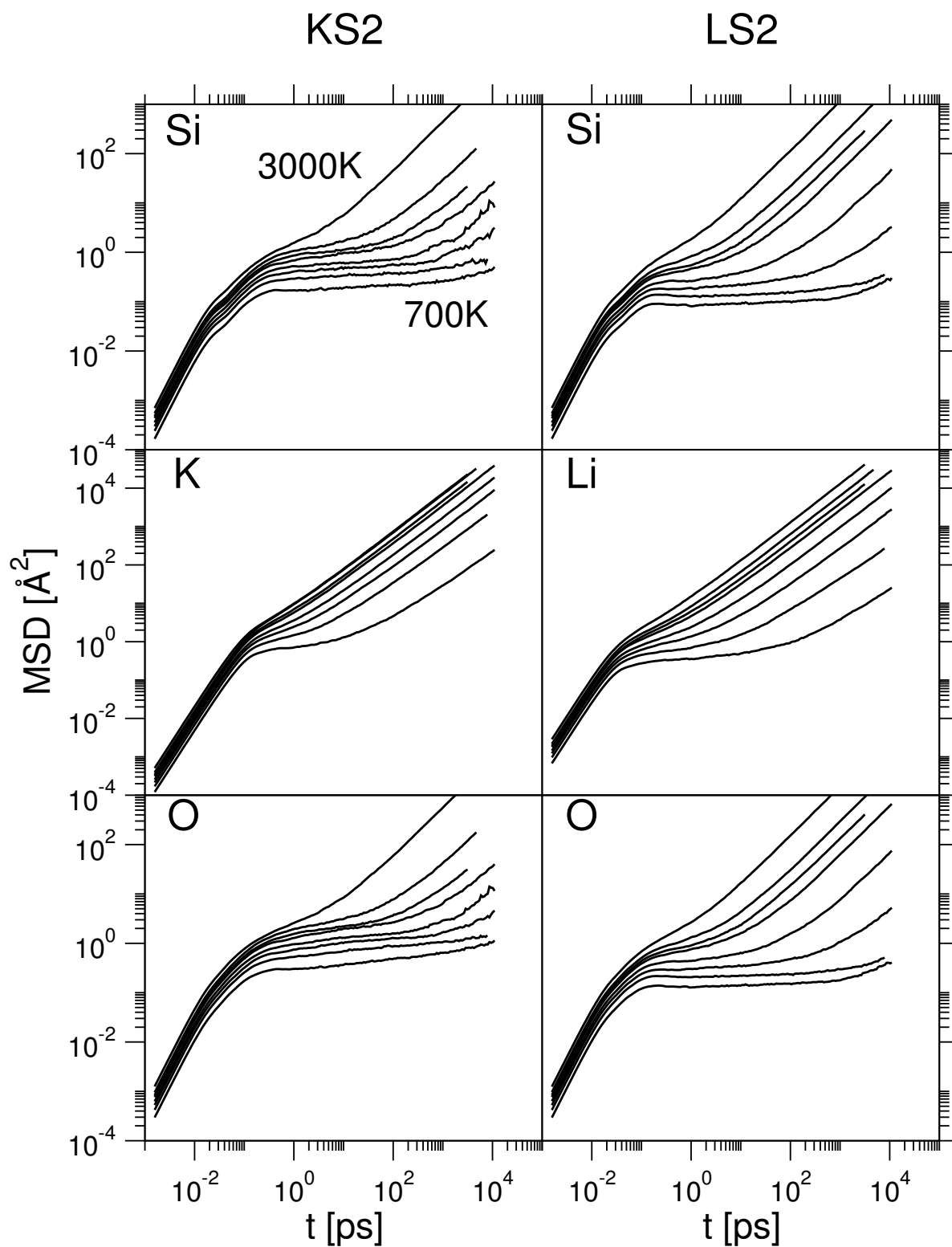
Verwendet man im diffusiven Bereich nun die Einstein–Relation, Gleichung (2.47), so erhält man die Selbstdiffusionskonstanten. Diese sind in Abbildung 3.16 als Arrheniusplot aufgetragen, also logarithmisch gegen die inverse Temperatur. Für die Matrixkomponenten erkennt man im Temperaturbereich zwischen 1500 K und 3000 K eine gute Übereinstimmung der Diffusionsdaten mit dem Arrheniusgesetz

$$D \propto \exp\left(-\frac{E_A}{k_B T}\right). \quad (3.1)$$

Dieses Verhalten ist aus anderen Simulationen zu SiO<sub>2</sub> bzw. Natriumsilikaten bekannt [6, 17]. Die dabei zugrundeliegende Vorstellung ist, daß die Teilchen sich in einer eingefrorenen Potentiallandschaft des Systems befinden, in der sie sich durch die Aktivierungsenergie  $E_A$  über eine Potentialbarriere in ein benachbartes Minimum begeben können. Man weiß, daß dieses Bild bei höheren Temperaturen nicht für die Matrixkomponenten anwendbar ist. Dort hat man dann auch kein Arrheniusverhalten der Selbstdiffusion. Das Arrheniusverhalten der Matrixkomponenten wird hier nur dadurch suggeriert, daß  $D_{Si}$  und  $D_O$  in Abbildung 3.16 nur über einen vergleichsweise kleinen Temperaturbereich geplottet sind. Man beobachtet hin zu höheren Temperaturen einen „Crossover“ in einen Bereich, in dem sich die Matrixkomponenten durch ein Potenzgesetz, wie von der Modenkopplungstheorie vorgeschlagen, beschreiben lassen (vgl. Kapitel 1.4, [6]).

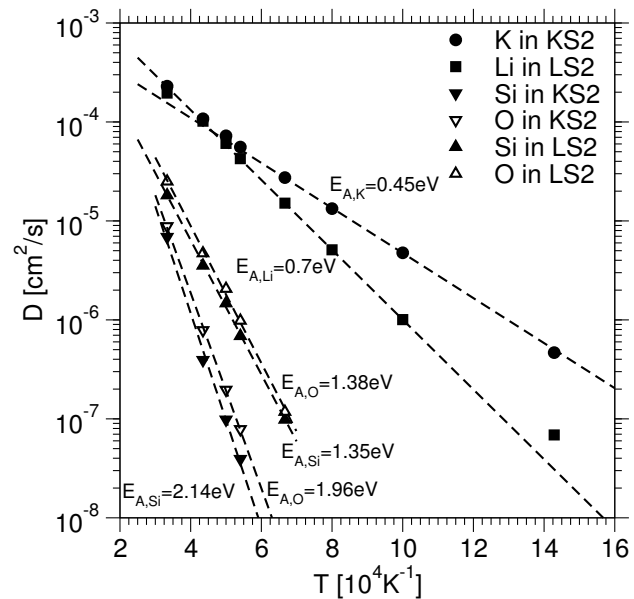
Generell erkennt man für die Alkaliionen deutlich niedrigere Aktivierungsenergien, als für die Matrixkomponenten. Das hat für fallende Temperaturen eine immer größer werdende Zeitskalentrennung zwischen diesen zur Folge. Speziell bei den niedrigsten Temperaturen, bei denen die Matrixkomponenten faktisch eingefroren sind, d.h. den Plateaubereich im Rahmen der Simulationszeit nicht mehr verlassen, kann man von einer völlig entkoppelten Dynamik von Alkaliionen und Matrixkomponenten sprechen.

<sup>14</sup>vgl. Kapitel 2.2.2.1.



**Abbildung 3.15.:** Übersicht der mittleren Verschiebungsquadrate (MSD) der binären Alkalidisilikate. Die gezeigten Temperaturen sind aufsteigend: 700 K, 1000 K, 1250 K, 1500 K, 1850 K, 2000 K, 2300 K und 3000 K.

**Abbildung 3.16.:** Arrheniusplot der Diffusionskonstanten für LS2 und KS2. Die gestrichelten Linien sind Fits des Arrheniusgesetzes (vgl. Gleichung (3.1)). Bei den Matrixkomponenten Si und O wurden die Fits über alle gezeigten Datenpunkte erzeugt, bei Li und K wurden nur die Temperaturen 1500 K, 1250 K und 1000 K verwendet. Die Übereinstimmung mit den höheren Temperaturen ist dabei gut. Man beobachtet jedoch bei Lithium eine Änderung der Aktivierungsenergie im Bereich zwischen 1000 K und 700 K.



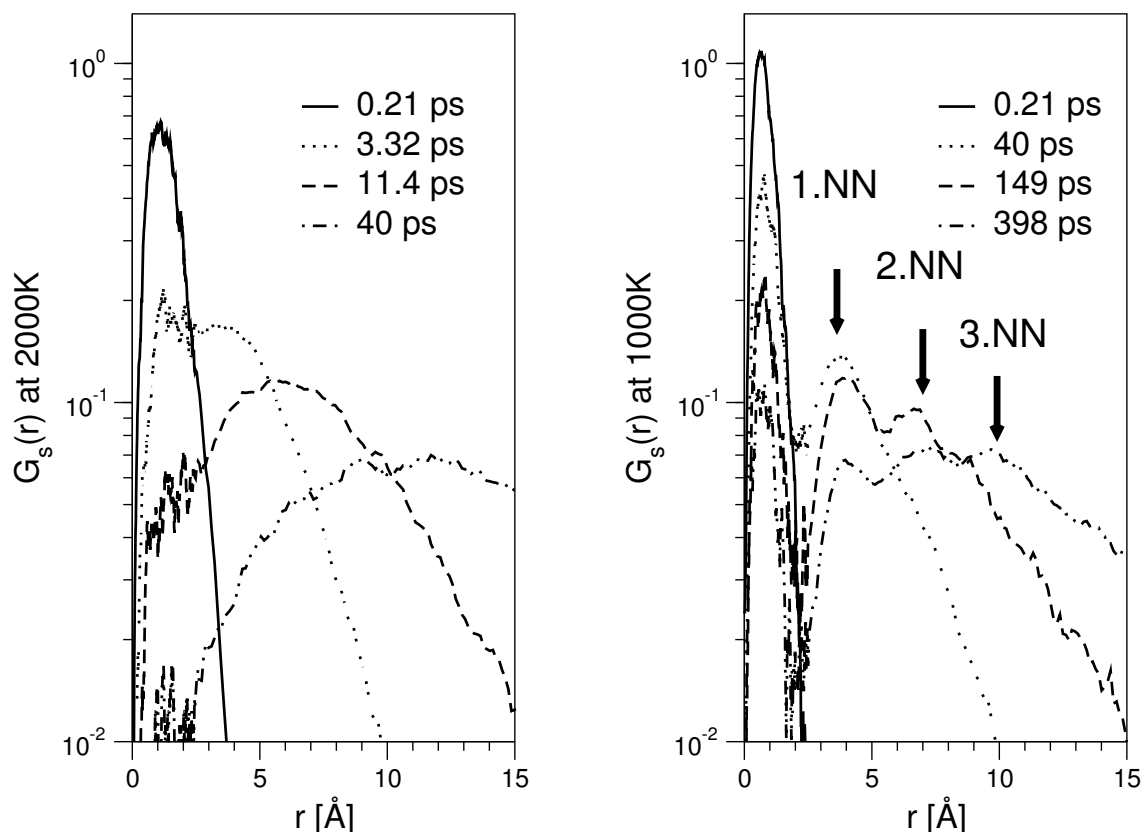
Dies spiegelt sich speziell auch in der Art wieder, wie sich die Alkaliionen fortbewegen. Eine exzellente Größe dies zu untersuchen, stellt der Selbstanteil der van-Hove-Korrelationsfunktion<sup>15</sup>  $G_s(r, t)$  dar. Diese ist proportional zu der Wahrscheinlichkeit, ein Teilchen zum Zeitpunkt  $t \neq 0$  im Abstand  $r$  von seinem Ort zum Zeitpunkt  $t = 0$  zu finden. Betrachtet man nun beispielsweise in Abbildung 3.17 den Plot für 2000 K, so erkennt man für sehr kurze Zeiten (hier 0.21 ps) noch eine Verteilung um die Ursprungsposition der Kaliumionen. Für größere Zeiten erhält man ausgeschmierte Verteilungen, deren Schwerpunkt sich vom Ursprung entfernt. Anders jedoch das Bild für niedrige Temperaturen, z.B. im Plot für 1000 K. Dort bilden sich mit fortschreitender Zeit neben dem Maximum um den Ursprung der Teilchen noch strukturierte Maxima, also ausgezeichnete und relativ feste Plätze innerhalb der Matrix, auf den Längenskalen der NN-Abstände. Das bedeutet, daß sich die Alkaliionen in einer eingefrorenen Matrix durch Sprünge auf Nachbarionenpositionen, sogenannte *Hopping*-Prozesse, in ihren Netzwerken fortbewegen.

Bei niedrigen Temperaturen und eingefrorener Matrix gibt es also scheinbar ausgezeichnete Plätze für die Alkaliionen. Das kann man gut anhand der kohärenten intermediären Streufunktionen<sup>16</sup> überprüfen. In Abbildung 3.18 finden sich die kohärenten ( $F(q, t)$ ) wie die inkohärenten intermediären Streufunktionen ( $F_s(q, t)$ ) für Kalium bzw. K-K in KS2 und Lithium bzw. Li-Li in LS2 bei 1000 K und 2000 K. In grau wurden zum Vergleich die Korrelationen der Matrix von Si bzw. Si-Si den Alkali-Alkali-Korrelationen gegenübergestellt. Die gewählten  $q$ -Vektoren liegen dabei nahe den Prepeaks  $q_1$ ,  $q_{2a}$  und  $q_{2b}$  sowie den  $q$ -Vektoren der NN-Alkali-Abstände (FSDP).

Betrachtet man bei 1000 K und 2000 K die  $F_s(q, t)$  und  $F(q, t)$  der Matrixkomponente Si, so sieht man, daß der Zerfall beider Korrelation fast aufeinander liegt, also auf der gleichen Zeitskala abläuft. Bei 1000 K verharren die Korrelationsfunktionen für große Zeiten auf dem

<sup>15</sup>vgl. Kapitel 2.2.2.2 zur Definition der van-Hove-Korrelationen.

<sup>16</sup>vgl. Kapitel 2.2.2.3 zu Streufunktionen.



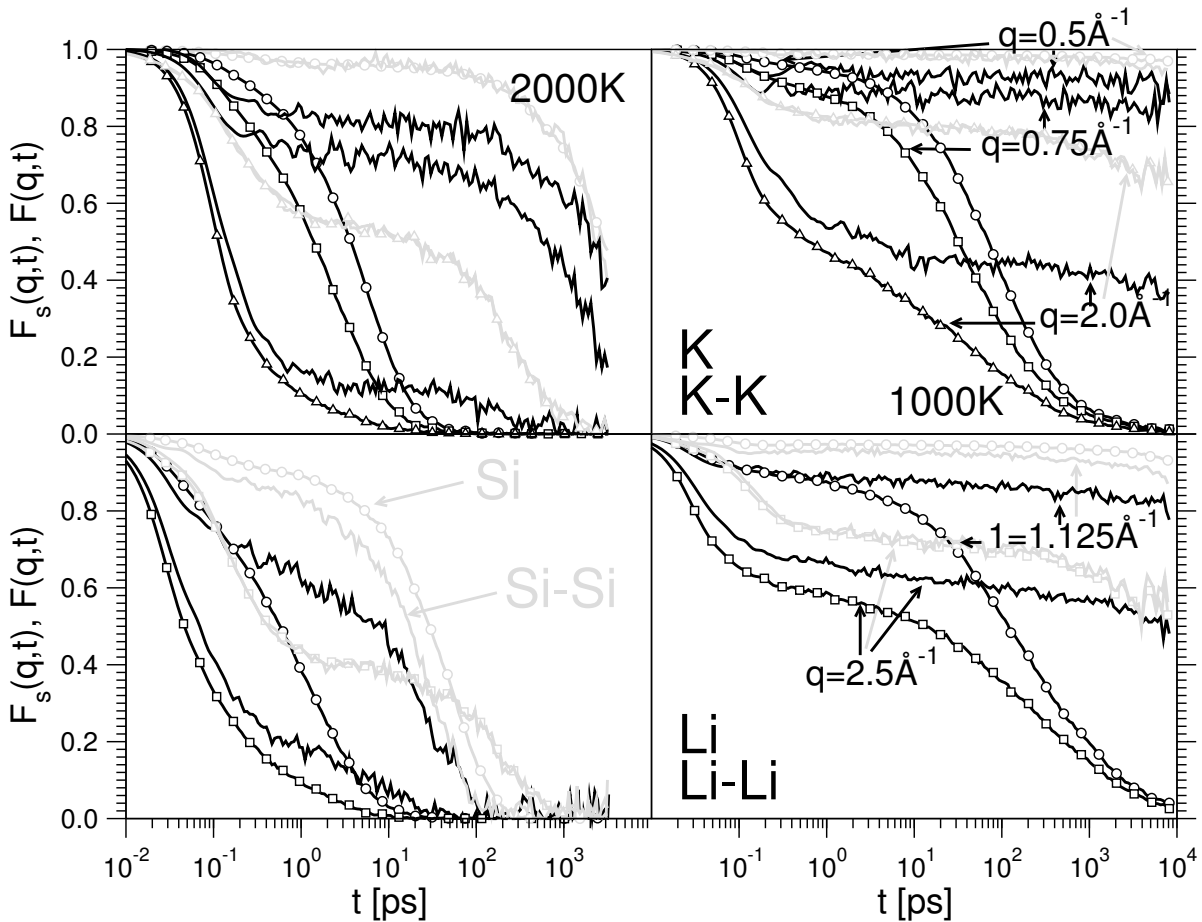
**Abbildung 3.17.:** Selbstanteil der van-Hove-Korrelationsfunktion für Kalium in KS2. Bei 2000 K sieht man für längere Zeiten keine Struktur mehr. Bei 1000 K jedoch bleibt der Peak an der Ausgangsposition der Teilchen erhalten. Zudem bilden sich an den Positionen der Nächste-Nachbar Peaks (NN) Maxima. Dazwischen findet man, speziell zwischen ersten und zweiten NN-Peak, Minima. Daran erkennt man, dass die Selbstdiffusion der Alkaliionen bei niedrigen Temperaturen (im Glas), durch Hüpfprozesse, sogenanntes *hopping*, getrieben wird. Das Verhalten der Lithiumionen in LS2 ist analog.

Plateau, die Matrix ist also eingefroren.<sup>17</sup> Anders ist das Verhalten für die Streufunktionen der Alkaliionen. Betrachtet man die  $F_s(q, t)$  bei 2000 K, so sieht man, daß diese ca. 2.5 Zeitgrößenordnungen schneller zerfallen, als die zum gleichen  $q$ -Vektor errechneten  $F(q, t)$ . Auch zu sehen ist, daß die Zeitskala, auf der die  $F(q, t)$  der Alkaliionen zerfallen, die gleiche ist, auf der die  $F(q, t)$  der Matrixkomponente Si zerfällt. Diese Beobachtung gilt auch für die kohärenten Korrelationen bei 1000 K. Nur daß bei dieser Temperatur gar kein Zerfall der  $F(q, t)$  in der Simulationszeit mehr festzustellen ist, wohingegen die zugehörigen  $F_s(q, t)$  um ca. 3 Zeitgrößenordnungen langsamer zerfallen, als bei 2000 K.

Was kann man daraus schließen: Man stellt fest, daß es schon in der Schmelze (2000 K), in der alle Komponenten noch im Gleichgewicht sind, eine Trennung zwischen der Zeitskala der Dynamik der Alkaliionen, die durch die  $F_s(q, t)$  ausgedrückt wird, und der Lebensdauer ausgezeichnete Plätze für die Alkaliionen gibt, repräsentiert durch die  $F(q, t)$ , die durch die Zeitskala der Relaxation der Matrix bestimmt wird. Friert die Matrix ein, so bleiben auch die ausgezeichnete Plätze erhalten, obwohl die Alkaliionen weiterhin ins diffusive Regime

<sup>17</sup>Warum der Plateauwert für große Zeiten dennoch langsam abfällt, wird in Kapitel 3.3 behandelt.



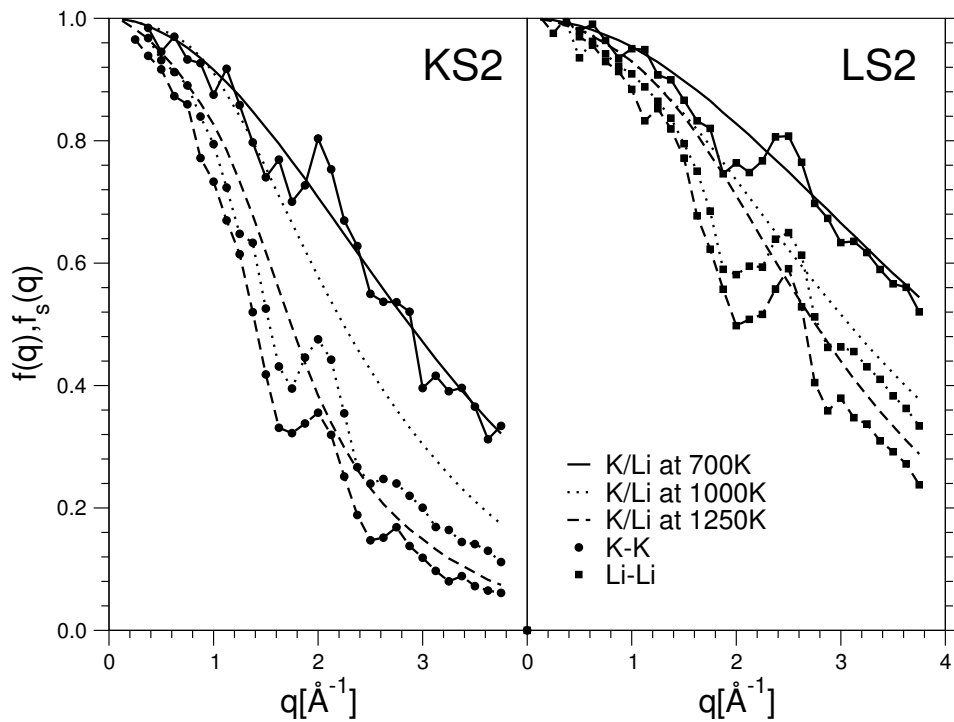


**Abbildung 3.18.:** Kohärente– (durchgezogene Linien) und inkohärente (Linien mit offenen Kreisen) intermediäre Streufunktionen für 2000 K (linke Spalte) und 1000 K (rechte Spalte), sowohl für Kalium in KS2 (oberer Reihe) und Lithium in LS2 (untere Reihe).  $q$ -Vektoren links entsprechen der rechts. Beschreibung im Text ab Seite 64.

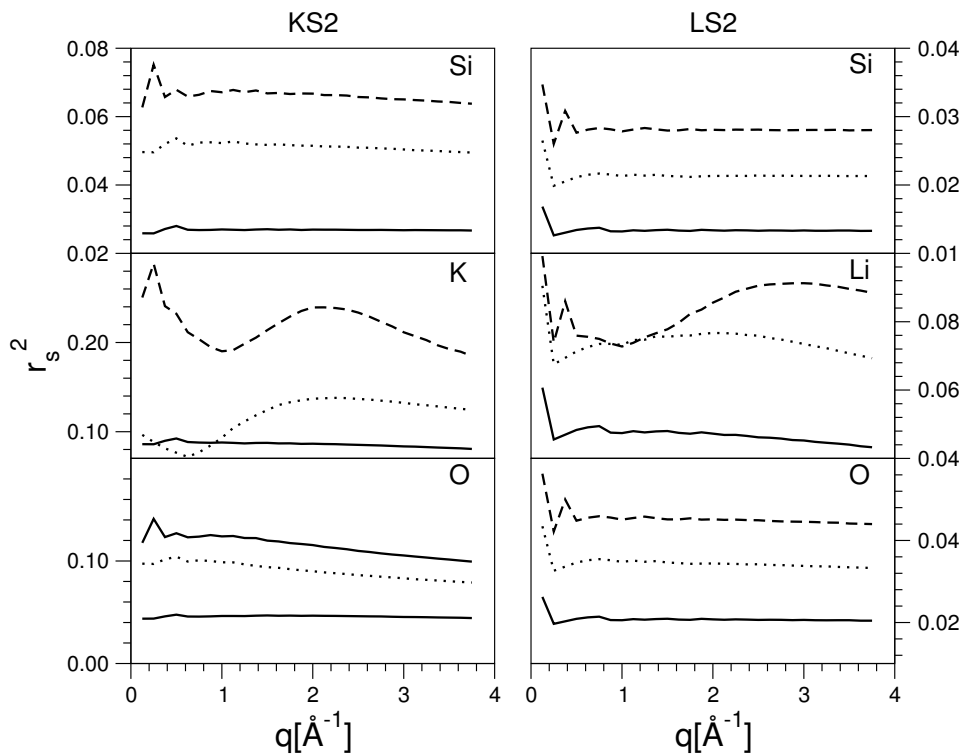
gelangen, wenn auch deutlich langsamer als in der Schmelze. Die Ionen bewegen sich also, wohingegen die Plätze, auf denen sie sich bewegen, fest in die Matrix eingebettet sind.

Bestimmt man nun mithilfe von Gleichung (1.11) die Plateauhöhen  $f_c(q)$  von  $F(q,t)$  und  $F_s(q,t)$ , so erhält man die  $q$ -abhängigen Debye–Waller– respektive Lamb–Mößbauer–Faktoren. Diese sind in Abbildung 3.19 für KS2 und LS2 für die drei niedrigsten Temperaturen in dieser Arbeit geplottet. Die Debye–Waller– und Lamb–Mößbauer–Faktoren repräsentieren ein Maß für die Lokalisierung von Plätzen bzw. Teilchen. Das bedeutet, man erkennt für alle Korrelationen, daß die Lokalisierung für fallende Temperaturen schärfer wird. Beim Vergleich zwischen KS2 und LS2 fällt zudem auf, daß bei gleicher Temperatur die Kaliumionen in KS2 schwächer lokalisiert sind, als es für Lithium in LS2 der Fall ist. Bei den Debye–Waller–Faktoren erkennt man neben dem  $q$ -abhängigen Abfall noch Struktur, die in Phase mit dem statischen Strukturfaktor ist. Sehr gut zu erkennen sind vor allem die NN-Abstände bei  $q \approx 2 \text{ \AA}^{-1}$  (KS2) und  $q \approx 2.5 \text{ \AA}^{-1}$  (LS2).

In guter Näherung können die  $f_s(q)$  durch eine Gaußfunktion beschrieben werden, wie man



**Abbildung 3.19.:**  $q$ -abhängige Debye-Waller- ( $f(q)$ , Linien mit Symbol) und Lamb-Mößbauer-Faktoren ( $f_s(q)$ , Linien).



**Abbildung 3.20.:** Quadrat der Lokalisierungslängen der Lamb-Mößbauer-Faktoren  $r_s^2$  für die Temperaturen 700 K (durchgezogene Linie), 1000 K (gepunktete Linie) und 1250 K (gestrichelte Linie).

schon in einigen Glasbildnern beobachtet hat, z.B. auch in  $\text{SiO}_2$  [59]. Wenn man  $f_s(q)$  als Gaußfunktion beschreiben kann, so existiert eine konstante Lokalisierungslänge  $r_s$ , so daß

$$f_s(q) = \exp(-r_s^2 q^2). \quad (3.2)$$

Beschreibt Gleichung (3.2)  $f_s(q)$ , so befindet man sich in der harmonischen Näherung. Weicht allerdings ein Fit der Form von Gleichung (3.2) von der tatsächlichen Beobachtung ab, wie es bereits in Simulationen zu NS2 beobachtet wurde (vgl. Abbildung 1.6, rechts), so enthält die Lösung Anharmonizitäten, d.h. es existiert nicht nur eine konstante Lokalisierungslänge  $r_s$ , sondern man hat es vielmehr mit einer  $q$ -abhängigen Lokalisierungslänge  $r_s(q)$  zu tun, die die Anharmonizität quantifiziert. Damit erhält man aus Gleichung (3.2) den Ansatz

$$r_s^2(q) = -\frac{1}{q^2} \ln f_s(q),$$

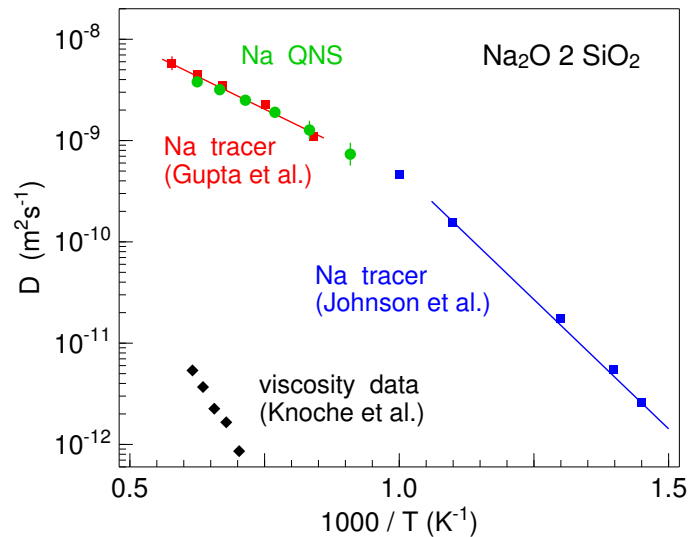
mit dem man die  $q$ -abhängigen Lokalisierungslängen bestimmen kann.

Diese sind in Abbildung 3.20 aufgetragen. Man erkennt, daß die Matrixkomponenten generell gut durch eine Konstante approximierbar sind. Größere Abweichungen sieht man nur für sehr kleine  $q$ . Eine weitere Abweichung ist speziell für Si in KS2 hin zu großen  $q$  erkennbar: Dort sieht man einen leichten Abfall von  $r_s^2$ . Von speziellem Interesse ist hier jedoch das Verhalten der  $r_s^2$  von Lithium und Kalium. Für die niedrigste Temperatur, 700 K, kann man  $r_s^2$  näherungsweise mit einer Konstanten beschreiben. Für höhere Temperaturen allerdings weist  $r_s^2$  für sowohl kleine als auch große  $q$  eine komplexe Struktur auf, d.h.  $r_s^2$  verhält sich stark anharmonisch und damit ist die Gaußsche Approximation, wie sie zuvor angesprochen wurde, für Alkaliionen nur bei niedrigen Temperaturen gültig. Das ist ein Verhalten, daß auch schon bei anderen Alkalisilikaten, vor allem für verschiedene Natriumsilikate [59] beobachtet wurde. Für die Matrixkomponenten kann man die Approximation noch bei höheren Temperaturen nutzen.

Die Dynamik verschiedener binärer Systeme zeigt wiederum viele qualitative Übereinstimmung zueinander. Grundsätzlich zeichnet sie sich durch eine Zeitskalentrennung zwischen Matrix und Alkaliionen aus. In der Matrix bewegen sich die Alkaliionen auf ausgezeichneten Plätzen, die eine Art perkolierendes Netzwerk durch die Matrix bilden. Die Plätze dieses Netzwerks zerfallen auf der Zeitskala der Matrix, d.h. fällt die Matrix aus dem Gleichgewicht, so „frieren“ auch die ausgezeichneten Alkaliplätze in der Matrix ein. Die Alkaliionen bewegen sich auch in diesen eingefrorenen Plätzen noch diffusiv fort.

Es wurde auch gezeigt, daß die Lamb-Mößbauer-Faktoren für die Matrixkomponenten sich gut harmonisch nähern lassen, diese Näherung allerdings nur für die Lamb-Mößbauer-Faktoren der Alkaliionen bei niedrigen Temperaturen gilt. Sind die Temperaturen etwas höher, so zeigt sich eine durch  $r_s(q)$  quantifizierbare Anharmonizität. Das geschilderte Verhalten der Lamb-Mößbauer-Faktoren stimmt gut überein mit bereits bekannten Resultaten aus Natriumsilikatsimulationen. Das ist wieder in Übereinstimmung mit qualitativen Eigenschaften verschiedener Alkaliionen in Alkalisilikat.

**Abbildung 3.21.:** Auftragung von verschiedenen Natriumdisilikatdiffusionsmessungen. Enthalten sind Viskositätsmessungen von Knoche et al. [127], quasielastische Neutronenstrahlungsergebnisse (QNS) von Kargl et al. [126] und Tracer-Diffusionsdaten von Gupta and King [128] und Johnson et al. [129]. Gut zu erkennen ist die Veränderung der Aktivierungsenergie von Natrium unterhalb ca. 1000 K. Die Aktivierungsenergie steigt für die untersuchten Temperaturen unter 1000 K. Zur Erklärung der Daten aus den Viskositätsmessungen, siehe Text ab Seite 69.



### 3.3. Alterungs-Phänomene

Geht eine Schmelze in ein Glas über, fällt sie also aus dem Gleichgewicht, so äußert sich das sowohl in dynamischen Größen, wie z.B. dem Verhalten der Diffusionskonstanten, als auch in statischen Größen, wie z.B. der isothermen Kompressibilität  $\kappa_T$  (vgl. z.B. [6], dort wird für  $\text{SiO}_2$  gezeigt, wie der Wert des totalen statischen Struktur factors  $S(q)$  für den Wellenvektor  $q \rightarrow 0$  aus der Flüssigkeit eingefroren wird, d.h. sich bei niedrigen Temperaturen kaum mehr verändert. Aus der Beziehung  $\lim_{q \rightarrow 0} S(q) = \rho k_B T \kappa_T$  [130], mit der Dichte  $\rho$  und der Temperatur  $T$ , erhält man  $\kappa_T$ ).

Bei Alkalisilikaten ist das Verhalten noch komplizierter, da selbst unterhalb der experimentellen Glasübergangstemperatur<sup>18</sup>  $T_{g,\text{exp}}$  die Alkaliionen im System noch mobil sind.<sup>19</sup> Denn wie in Kapitel 3.2 erwähnt, besteht eine Zeitskalentrennung der Mobilität zwischen den „langsamen“ Matrixkomponenten und den „schnellen“ Alkaliionen. Damit stellt sich die Frage, ob und wie die Matrix durch ihr Einfrieren einen Einfluß auf die Alkaliionendiffusion hat.

Zunächst werden in Abbildung 3.21 experimentelle Daten von Systemen betrachtet, die sich alle im Gleichgewicht befinden. Auffällig sind zunächst in Abbildung 3.21 die aus den Viskositätsdaten abgeschätzten Diffusionskonstanten. Letztere wurden mittels der Stokes-Einstein-Relation,

$$D = \frac{k_B T}{\eta \lambda},$$

bestimmt, wobei  $\lambda$  in der phänomenologischen Eyring-Theorie [131] die Bedeutung der Länge eines elementaren Diffusionssprunges hat.<sup>20</sup> Es sei betont, daß die Stokes-Einstein-Beziehung in diesem Zusammenhang rein phänomenologisch ist und in vorliegendem Fall eine

<sup>18</sup>zur Definition der experimentellen Glasübergangstemperatur, siehe Kapitel 1.1.

<sup>19</sup>Für Natriumdisilikat beträgt  $T_{G,\text{exp}} = 740 \text{ K}$  [127].

<sup>20</sup>Es wurde aber, unter anderem für  $\text{SiO}_2$ , gezeigt, daß  $\lambda$  nicht immer konstant ist. Das schränkt die Anwendbarkeit des Stokes-Einstein-Gesetz ein (vgl. [6]).

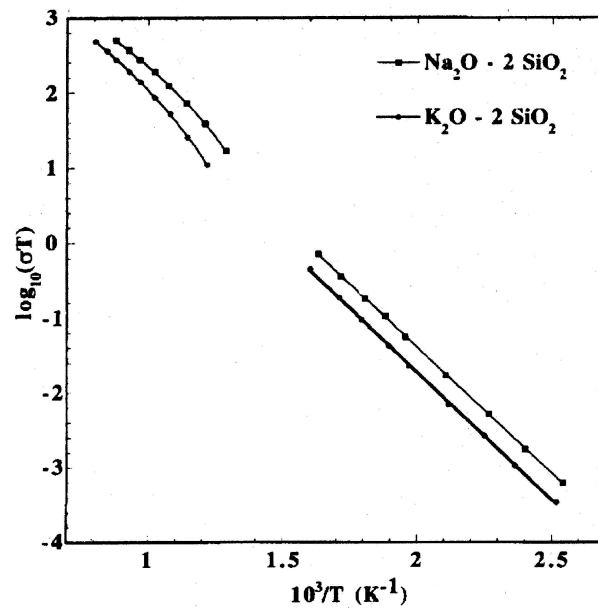


Abbildung 3.22.: Experimentelle Leitfähigkeitsmessungen zu NS2 und KS2 [133].

grobe Abschätzung einer mittleren Diffusionskonstante für Si und O, d.h. die langsamen Matrixkomponenten, liefert.

Die anderen aufgetragenen Messungen zeigen Na-Diffusionskonstanten aus Tracer-Diffusionsmessungen und quasielastischer Neutronenstreuung (QNS). Erwähnenswert dabei ist, daß im Bereich hoher Temperaturen, in denen experimentelle Messungen schwieriger werden, die Daten aus den Tracer-Diffusionsmessungen und der inelastischen Neutronenstreuung so konsistent sind. Dies bekräftigt die Richtigkeit der Ergebnisse, da die Resultate zweier völlig verschiedener Meßmethoden sich gegenseitig bestätigen.

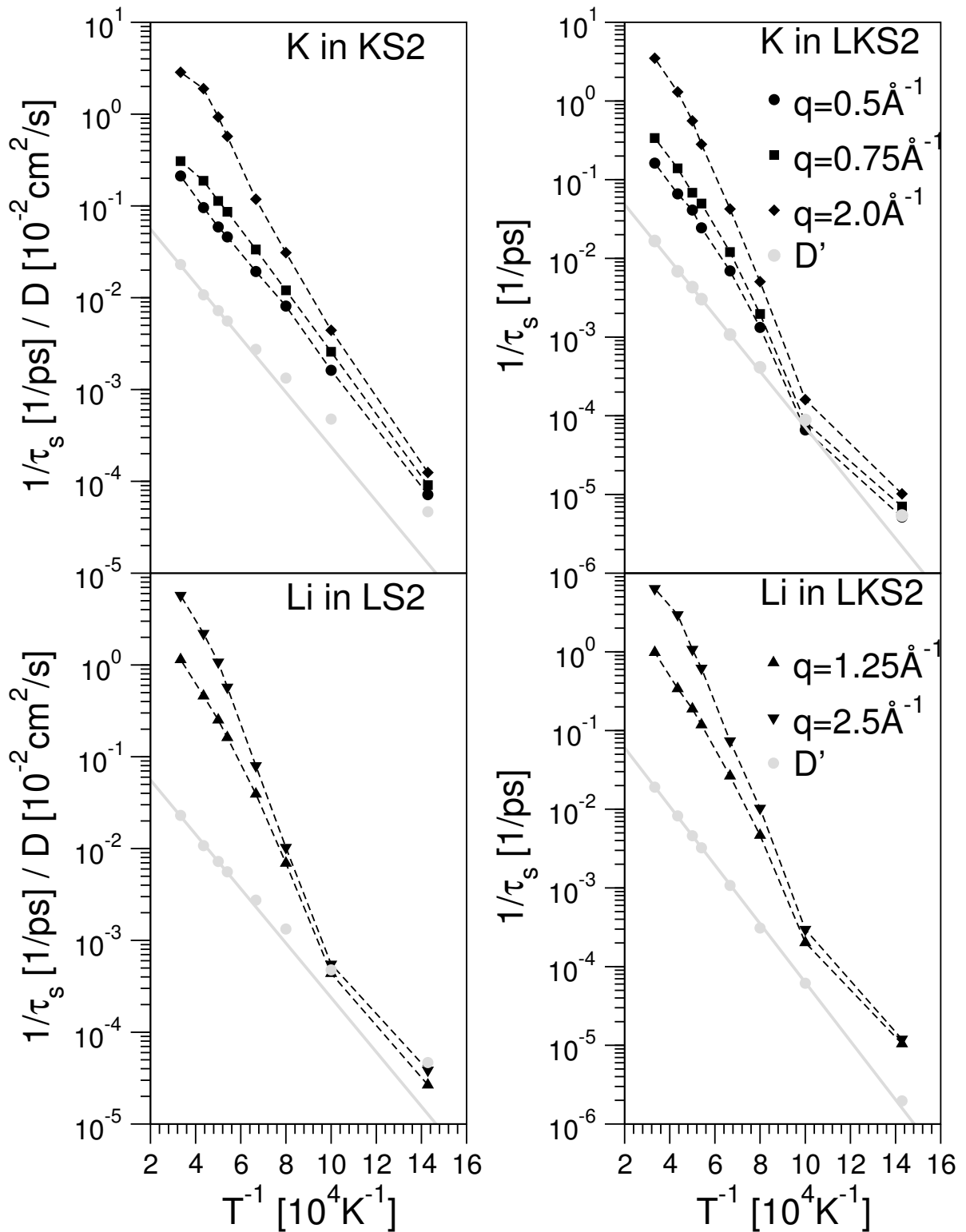
Die Diffusionskonstanten erhält man in Neutronenstreuexperimenten indirekt über die inkohärente Streufunktion. Durch die inkohärenten Streufunktionen kann man die  $q$ -abhängigen Relaxationszeiten  $\tau(q)$  bestimmen und dann unter Zuhilfenahme der Relation (vgl z. B. [132])

$$\tau^{-1}(q) = Dq^2,$$

die im hydrodynamischen Limes, also für  $(q, \omega) \rightarrow 0$ , gilt, die Selbstdiffusionskonstante  $D$  bestimmen. Im Experiment wurden dabei  $q$ -Vektoren im Bereich von  $0.5 \text{ \AA}^{-1}$  bis  $1.2 \text{ \AA}^{-1}$  verwendet.

Weitere experimentelle Ergebnisse von Souquet and Lèvy [133] sieht man in Abbildung 3.22. Sie führten Leitfähigkeitsmessungen von NS2 und KS2 durch. Wie man bei NS2 gut im Vergleich zu den Daten in Abbildung 3.21 sehen kann, stimmt das qualitative Verhalten der Messungen in den gleichen Temperaturbereichen der Abbildungen gut überein. Zudem sieht man in Abbildung 3.22 ein sehr ähnliches Verhalten von NS2 und KS2.

Bei Betrachtung von Abbildung 3.21 und Abbildung 3.22 stellt man folgendes fest: Man sieht zwei Temperaturbereiche, in denen sich der Verlauf der Diffusionskonstanten bzw. der Leitfähigkeiten verändert. Einmal erkennt man bei ca. 1000 K ein konkaves Abknicken der Verläufe, desweiteren dann bei ca. 700 K eine konvexe Änderung der Verläufe. Die Änderung bei



**Abbildung 3.23.:** Arrheniusplot der inversen Relaxationszeiten der inkohärenten Streufunktionen  $1/\tau_s$ , aufgetragen im Vergleich zu den Diffusionskonstanten  $D$ . Während die Diffusion (vor allem in LKS2) über den ganzen Temperaturbereich ein Arrheniusverhalten aufweist (durchgezogene graue Linien), ist in den Relaxationszeiten eine deutliche Veränderung im Verhalten zu erkennen (siehe Text ab Seite 72). Vorweggenommen sind hier Resultate für LKS2.

ca. 700 K fällt in den Bereich der Temperatur des Glasüberganges  $T_g$ . Beim Vergleich der experimentellen Daten mit denen aus der Simulation (z.B. den Diffusionskonstanten aus Abbildung 3.23) fällt auf, daß die Simulationsdaten über den gesamten betrachteten Bereich ein Arrheniusverhalten aufweisen, ganz im Gegensatz zu eben besprochenen experimentellen Resultaten. Daher stellt sich die Frage, ob vielleicht das verwendete Potential<sup>21</sup> nicht in der Lage ist, das Verhalten in dem Temperaturbereich richtig zu reproduzieren. Dieser Annahme steht entgegen, daß Horbach [6, 59] mit einem anderen Potential<sup>22</sup> für NS2 die gleichen Beobachtungen zum Verlauf der Diffusionskonstanten machte, wie sie in dieser Arbeit für LS2 und KS2 gemacht werden. Das legt die Vermutung nahe, daß das in der Simulation beobachtete Arrheniusverhalten nicht eine Schwäche der verwendeten Potentiale ist, sondern auf einem anderen Unterschied zwischen Experiment und Simulation beruht.

Nimmt man jetzt die MSD's aus Abbildung 3.15, dann kann man erkennen, daß innerhalb des Simulationszeitraums die Matrixkomponenten bei Temperaturen unterhalb 1500 K definitiv nicht in den diffusiven Bereich vorgedrungen sind, also sich noch nicht wirklich vom Plateau entfernt haben. Ein besserer Indikator, ob sich eine Teilchensorte nicht im Gleichgewicht befindet, sind die inkohärenten Streufunktionen  $F_s(q, t)$ . Sind diese nicht auf 0 zerfallen, so bestehen noch Korrelationen, also kann man bei der Teilchensorte nicht davon ausgehen, daß sie sich im Gleichgewicht befindet. Die langsamste Teilchensorte im System ist das Silizium. Daher wurden die  $F_s(q, t)$  von Silizium in der Nähe des *First-Sharp-Diffraction-Peaks* (FSDP)<sup>23</sup> betrachtet.<sup>24</sup> Bei KS2 fällt die Matrix im Bereich  $< 1500$  K aus dem Gleichgewicht, bei LS2 schon für Temperaturen  $< 1850$  K. Wie aber eben erwähnt, sieht man erst bei 700 K, und nur in LS2, eine Veränderung der diffusiven Dynamik. Die Selbstdiffusion der Alkaliionen scheint zunächst nicht von der Veränderung der Dynamik der Matrix beeinflußt zu sein, nachdem sie schon weit oberhalb und auch danach noch ein unverändertes Arrheniusverhalten zeigt.

Anhand dieser Ergebnisse, könnte man die Vermutung aufstellen, daß die Matrix nur einen geringen Einfluß auf die Dynamik der Alkaliionen hat. Betrachtet man jedoch die Relaxationszeiten<sup>25</sup> der  $F_s(q, t)$  und vergleicht sie mit den Diffusionskonstanten der in dieser Arbeit untersuchten binären und ternären Systeme (Abbildung 3.23), so erkennt man, daß in den Relaxationszeiten  $\tau_s$  sehr wohl eine Veränderung bei Temperaturen zwischen 1500 K und 1250 K eintritt. Bei logarithmischer Auftragung verläßt der Verlauf der  $\tau_s$  in diesem Bereich die konkave funktionale Form.<sup>26</sup>

Für dieses widersprüchliche Verhalten, daß man einerseits in den Diffusionskonstanten keinen Einfluß der Veränderungen in der Matrix sieht, andererseits die Relaxationszeiten  $\tau_s$  ihren Verlauf ändern, findet man die Erklärung in der unterschiedlichen Sensitivität der beiden Größen auf Längenskalen. Die Diffusion ist ein Prozeß, der auf großen Längenskalen ( $q \rightarrow 0$ )

---

<sup>21</sup>vgl. Kapitel 2.1.2.1.

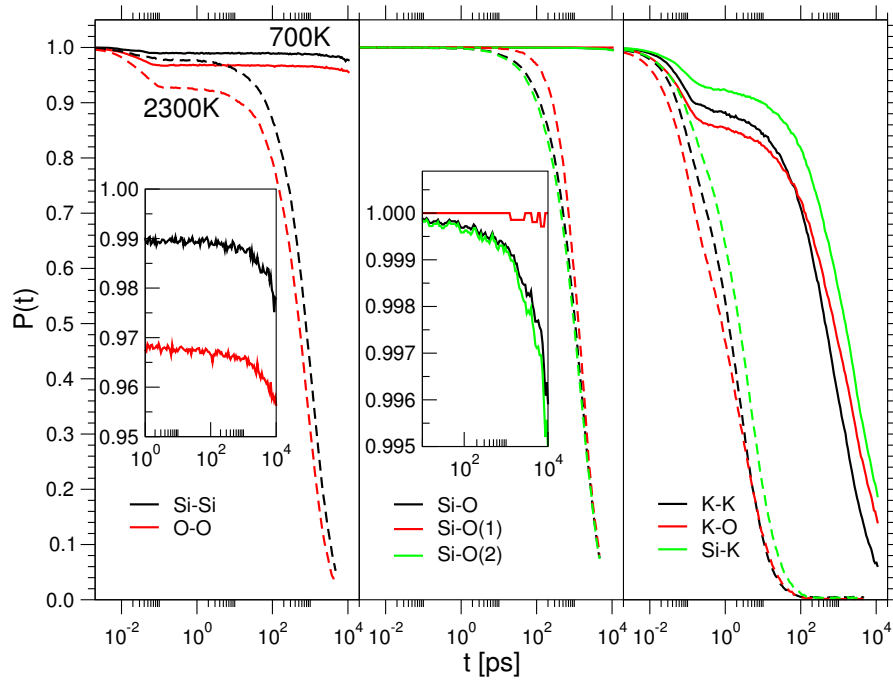
<sup>22</sup>Es handelt sich dabei um das BKS-Potential (vgl. dazu Infokasten auf Seite 34).

<sup>23</sup>Der FSDP ist im  $q$ -Raum der entsprechende Peak zum ersten Maximum der Paarkorrelation im Realraum.

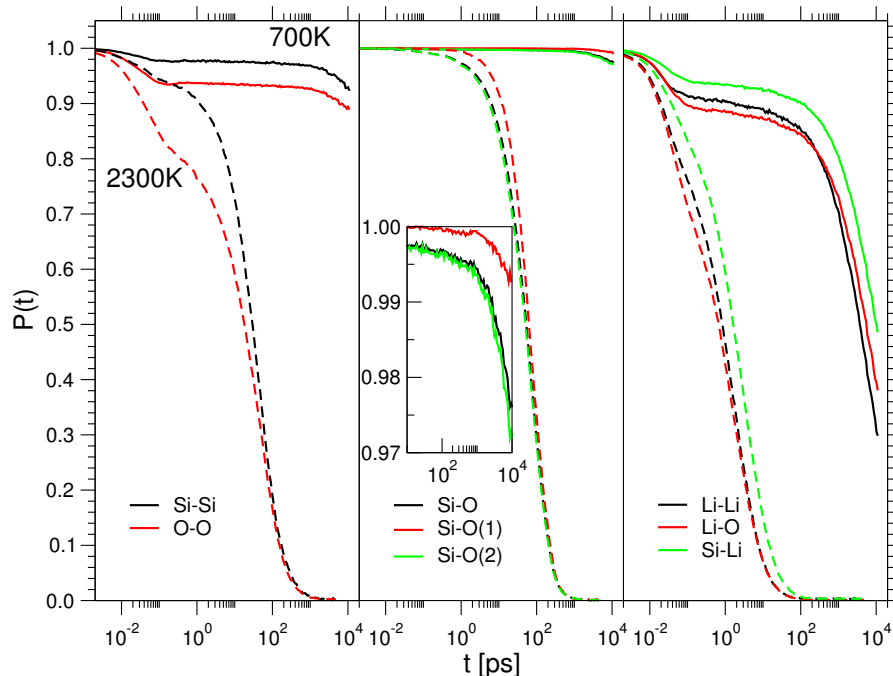
<sup>24</sup>Die einzelnen  $F_s(q, t)$  werden hier nicht im Detail gezeigt.

<sup>25</sup>Um die Relaxationszeiten der  $F_s(q, t)$  zu bestimmen, wurde der Zeitpunkt ausgelesen, an dem die Korrelation auf  $\frac{1}{10}$  abgefallen ist. Ist die Simulation (bei niedrigen Temperaturen) nicht lange genug gewesen, um diesen Punkt zu erreichen, so wurde der Zeitpunkt mithilfe eines Kohlrusch-Fits bestimmt (vgl. Kapitel 1.4).

<sup>26</sup>Untersuchungen zu Natriumdisilikat [59] zeigen, daß die  $\tau_s$  für hohe Temperaturen einen funktionalen Verlauf nach dem Potenzgesetz der MCT aufweisen, im Gegensatz zu den Diffusionskonstanten, die im selben Temperaturbereich ein Arrheniusverhalten zeigen.

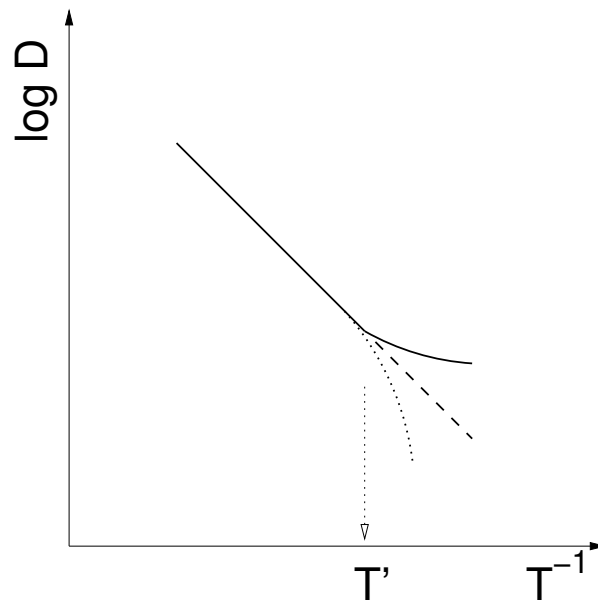


**Abbildung 3.24.:** Partielle Bond-Lebensdauer-Wahrscheinlichkeiten über Zeit für KS2. In den Insets werden Teile der 700 K Daten zur besseren Ansicht vergrößert dargestellt. O(1) steht für Sauerstoffe, die einfach mit Si koordiniert sind. Entsprechend ist O(2) zweifach mit Si koordiniert.



**Abbildung 3.25.:** Partielle Bond-Lebensdauer-Wahrscheinlichkeiten über Zeit für LS2. Insets wie Abbildung 3.25





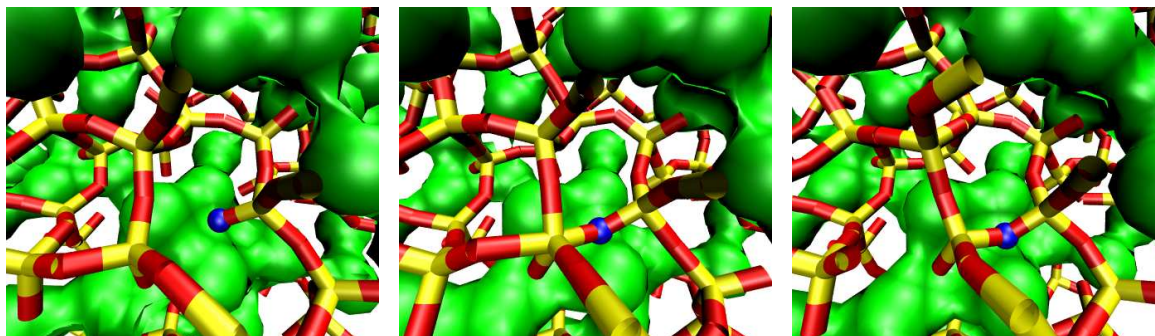
**Abbildung 3.26.:** Skizze zur Erklärung des Arrheniusverhaltens in der Simulation. Vergleiche dazu Text ab Seite 74.

abläuft. Bei dem Einfrieren der Dynamik der Matrix handelt es sich aber um einen Prozeß auf verschiedenen (endlichen) Längenskalen. Solche Veränderungen lassen sich gut mit  $F_s(q, t)$  erfassen, weshalb die aus  $F_s(q, t)$  hergeleitete Größe  $\tau_s$  die Veränderung der Matrixdynamik besser widerspiegelt, als die Diffusionskonstanten.

Es gibt aber auch eine weitere Erklärung, warum kein Einfluß der Matrixdynamik im Verhalten der Diffusionskonstanten der Alkaliionen zu sehen ist. Dazu betrachtet man die Skizze in Abbildung 3.26. Eingezeichnet ist hier der fiktive Verlauf der Diffusion in einem Arrheniusplot. Für  $T^{-1} < T'$  befindet sich das System noch im Gleichgewicht. Bei  $T'$  fällt die Matrix in der Simulation aus dem Gleichgewicht. Der gestrichelte Verlauf für  $T^{-1} > T'$  ist das Verhalten, das man aus dem Experiment kennt (vgl. z.B. Abbildung 3.21), wobei dort aufgrund der vergleichsweise großen zugänglichen Zeitskala<sup>27</sup> alle Komponenten im Gleichgewicht sind. Die konvexe Fortsetzung der durchgezogenen Linie rechts von  $T'$  ist das Verhalten, das man für die Diffusionskonstanten der Alkaliionen erwartet, wenn diese nicht im Gleichgewicht sind. Nun erhält man in der Simulation einen Verlauf der Diffusionskonstanten, der ein Arrheniusverhalten (gestrichelte Linie) auch unterhalb von  $T'$  suggeriert. Das kann aber einfach Folge davon sein, daß nicht alle Komponenten im Gleichgewicht sind, somit sich ein Verhalten zwischen den beiden genannten Verläufen ausbildet.

Aber selbst die Matrix, die offensichtlich aus dem Gleichgewicht gefallen ist, zeigt sich als nicht total eingefroren. Man beobachtet noch sogenannte Alterungs-Effekte, die man auch in den  $F(q, t)$  für die Matrix-Korrelationen findet. Das bedeutet, man erkennt für Si-Si bei 1000 K in Abbildung 3.18, daß das Plateau mit fortschreitender Zeit absinkt, ein Verhalten, das auch schon in anderen Alkalisilikaten festgestellt wurde [59]. Ein entsprechendes Verhalten

<sup>27</sup>Dabei wird hier von Zeitskalen gesprochen, die für die Simulation, die sich auf Zeitskalen der Größenordnung von Nanosekunden abspielt, unerreichbar sind.



**Abbildung 3.27.:** Drei „Schnappschüsse“ aus den Konfigurationen von KS2 bei 1000 K. In grün die Kaliumionen, dargestellt wie auf Seite 59 beschrieben. Die Matrixkomponenten Si und O sind mittels der Darstellung *Dynamic–Bonds* von VMD [125] dargestellt. Das bedeutet, die Bindungen zwischen den Atomen werden als zweifarbige Zylinder dargestellt. Dabei zeigen die Farben an, welches Atom sich an welchem Ende der Bindung befindet. Gelb sind hier die Si–Atome, O ist rot dargestellt. Der blaue Ball ist ein markiertes Sauerstoffteilchen. Eine Beschreibung des hier gezeigten Ereignisses findet sich im Text ab Seite 75.

findet man auch bei der Betrachtung der partiellen Bond–Lebensdauer.

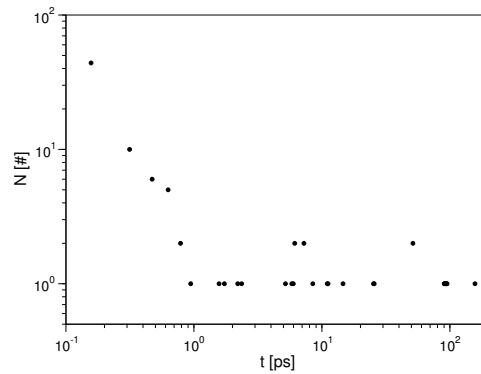
Ausgewählte Bond–Lebensdauern sind in Abbildung 3.24 und Abbildung 3.25 für eine hohe (2300 K) und eine tiefe Temperatur (700 K) dargestellt. Die Bond–Lebensdauer gibt dabei an, wie hoch die Wahrscheinlichkeit  $P(t)$  ist, eine zur Zeit  $t = 0$  existierende Bindung zur Zeit  $t$  noch vorzufinden. Eine Bindung ist dabei durch das erste Minimum der partiellen Paarkorrelationsfunktion definiert, d.h. es existiert eine Bindung, falls der Abstand  $r$  kleiner als die Position  $r_{\min,1}$  des ersten Minimum von  $g_{\alpha\beta}(r)$  ist.

Wie nicht anders zu erwarten, brechen bei der hohen Temperatur (fast) alle Bindungen in der Simulationszeit auf, d.h.  $P(t)$  zerfällt auf 0. Dabei zerfallen alle Korrelationen, bei denen mindestens eine Matrixkomponente beteiligt ist, deutlich langsamer als Korrelationen unter Beteiligung von Alkaliionen. Dies ist direkt auf die verschiedenen Aktivierungsenergien und die daraus resultierende Zeitskalentrennung zurückzuführen.<sup>28</sup>

Von großem Interesse ist hier jedoch das Verhalten bei der tiefen Temperatur. Dort fällt auf, daß ähnlich wie bei der kohärenten Streufunktion zuvor, die Bindungen zwischen Matrixkomponenten, also Bindungen zwischen Si–Si, Si–O und O–O, einen leichten Abfall für große Zeiten aufweisen (siehe Insets). Die Frage, die sich nun stellt, ist, welcher Mechanismus für diesen Zerfall verantwortlich ist?

In Abbildung 3.27 findet sich ein Beispiel, wie die Matrix sich selbst bei sehr niedrigen Temperaturen (hier 1000 K), noch verändert. Im ersten Bild (links), erkennen wir die Matrix, dargestellt wie in Abbildung 3.27 beschrieben. Blau markiert ist dabei das Sauerstoffatom, das gleich eine neue Bindung mit dem Si–Atom links von ihm eingehen wird. Im mittleren Bild erkennt man, daß sich speziell der links von dem markierten Teilchen gelegene Teil der Matrix derart deformiert hat, daß eine Bindung mit dem markierten Teilchen möglich wird. Für kurze Zeit ist das daran beteiligte Si–Atom 5–fach mit Sauerstoff koordiniert. Da diese Bindungszahl energetisch sehr ungünstig ist, bricht die Bindung zu einem der vorher gebundenen

<sup>28</sup>vgl. Kapitel 3.2.

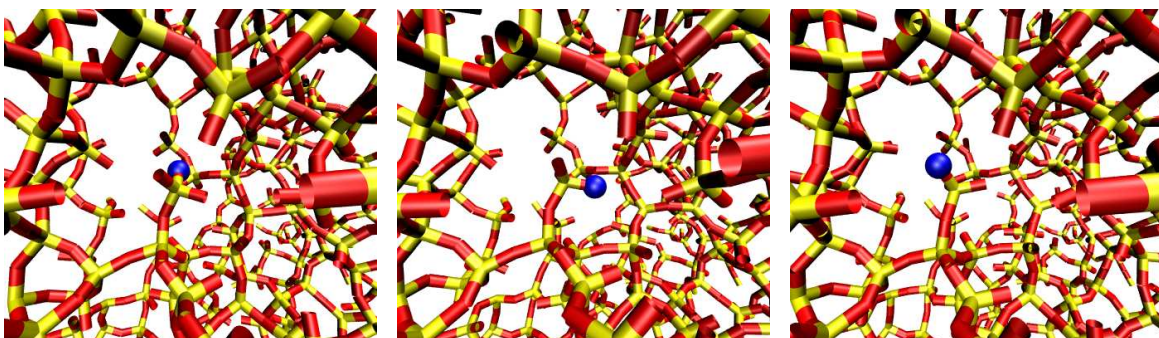


**Abbildung 3.28.:** Anzahlverteilung der Dauer der Defekte in der Matrix.

O-Atome schon im nächsten Schritt (rechtes Bild) wieder auf. Das Si-Atom ist nun mit dem markierten Sauerstoffatom gebunden, die lokale Struktur der Matrix hat sich verändert.

Der ganze Vorgang, der hier in drei Konfigurationsbildern dargestellt wird, dauerte ca. 0.3 ps. Diese Umordnung der Matrix wird nicht lange anhalten. Schon nach weiteren ca. 2.2 ps bricht die Bindung mit dem markierten Teilchen wieder auf, das Si-Atom geht wieder seine ursprünglichen Bindungen (linkes Bild) ein. In dieser Zeit relaxieren aber sowohl die umliegende Matrix, als auch die umliegenden Alkaliionen. Die Alkaliionen bewegen sich aufgrund der Zeitskalentrennung schneller, können sich also entsprechend schnell auf eine veränderte Matrixtopologie einstellen. Die umliegende Matrix erhält durch die eben beschriebene Veränderung neue Möglichkeiten, sich durch Streckung, Stauchung und Torsion umzuordnen.

Durch die Veränderung der Matrix ändert sich für diese ja auch die Potentiallandschaft, in der sie sich durch „Hopping“ fortbewegen. Das wiederum bedeutet aber, daß sich den Ionen hier ein zusätzlicher Diffusionskanal in der Art einer Defektdiffusion eröffnet. Die große Bedeutung dieser Beobachtung liegt darin, daß dies einen eminenten Unterschied zum Experiment darstellt. Dort relaxieren Glasmaterialien auf makroskopischen Zeitskalen, d.h. Spannungen in der Matrix und dadurch entstehende Defekte haben Zeit auszuheilen. Das bedeutet möglicherweise, daß im Experiment gemessene Diffusionskonstanten durchaus niedriger sind, als die in



**Abbildung 3.29.:** Weitere drei „Schnappschüsse“ aus den Konfigurationen von KS2 bei 1000 K. Darstellung wie in Abbildung 3.27, allerdings zur besseren Übersichtlichkeit ohne Alkaliionen. Gut zu erkennen: Das zentrale Tetraeder mit dem markierten Sauerstoff rotiert im Verlauf der Simulation um seine untere Si-O-Si-Bindung.

Simulationen ermittelten, da in Simulationen nicht lange genug äquilibriert werden kann, um alle Defektquellen auszuheilen. Daß ein Unterschied im Verhalten der Diffusionskonstanten besteht, sieht man alleine schon im Verhalten von NS2 bei 1000 K in Abbildung 3.21. Die Veränderung der Aktivierungsenergie, die man dort im Experiment im Gleichgewicht sieht, werden bei den simulierten Systemen in dieser Arbeit, aber auch in simulierten Systemen zu NS2 [59] nicht gesehen.

Um ein Gefühl dafür zu bekommen, wie langlebig die eben beschriebenen Defekte sind, wurde in Abbildung 3.28 ein nicht normiertes Histogramm der Zeiten aufgenommen, die nötig sind, daß ein einfach mit Si koordiniertes O-Atom, welches sich kurzfristig mit zwei Si-Atomen koordiniert, wieder nur einfach koordiniert ist und andersherum. Dazu wurde eine zeitlich aufgelöste Koordinationszahlenverteilung<sup>29</sup> aufgenommen und die Wechsel zwischen einfacher und mehrfacher Koordination und dem Wechsel zurück zur einfachen Koordination von O mit Si protokolliert. Es ist gut erkennbar, daß solche Koordinationswechsel einen kurzlebigen Charakter aufweisen mit typischen Koordinationsrückkehrzeiten von  $\lesssim 1$  ps. Der Anteil der Si-Atome, der an solchen Koordinationszahlwechseln beteiligt ist, liegt bei ca. 5 %. Also handelt es sich weder um häufige noch um dauerhafte Umbildungen der Matrix.

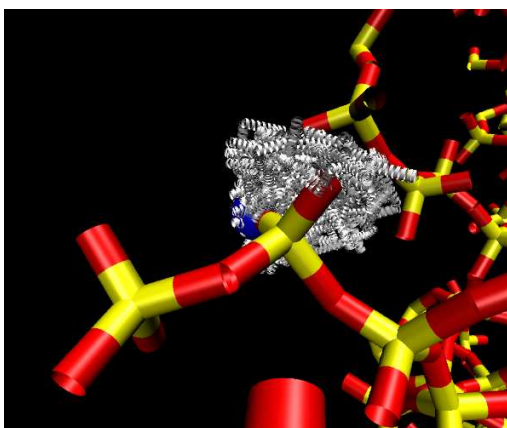
Wichtig ist auch die Frage, ob die Defekte wieder so ausheilen, daß die Teilchen wieder zu den Orten zurückkehren, die sie zur Zeit  $t = 0$  innehatten, oder ob eine Umordnung der Matrixkomponenten, vor allem von Sauerstoff, stattfindet. Um der Frage auf den Grund zu gehen, wurden für die einzelnen Sauerstoffatome die Squared-Displacements<sup>30</sup> (SQD) bestimmt. Dann wurden diejenigen Teilchen genauer untersucht, deren SQD größer ist, als das Quadrat des NN-Abstands von O-O.

Beispiele für solche Teilchen finden sich in den Abbildungen 3.27, 3.29 und 3.30. Speziell das markierte Teilchen in Abbildung 3.30 weist ein  $\text{SQD} > 30 \text{ \AA}^2$  auf.

Das Hauptaugenmerk soll aber jetzt auf das O-Atom der Abbildung 3.29 gelegt werden. Dort erkennt man im Zentrum des Bildes ein Matrixtetraeder, das nur an einem Sauerstoff mit der restlichen Matrix verbunden ist. Das bedeutet, das Tetraeder hat die Möglichkeit um diese Si-O-Si-Bindung zu rotieren, was tatsächlich auch geschieht. In Abbildung 3.31 ist diese Situation aus einem etwas anderen Blickwinkel dargestellt. Im rechten Teil der Abbildung

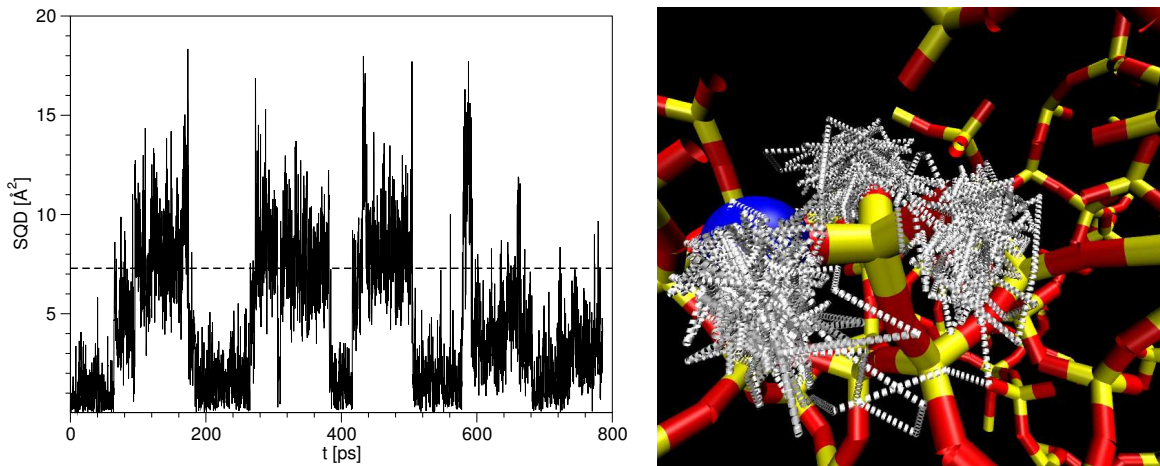
<sup>29</sup>siehe Kapitel 2.2.1.2.

<sup>30</sup>Das SQD eines Teilchens wird über  $r^2(t) = |\vec{r}_i(t) - \vec{r}_i(0)|^2$  bestimmt.



**Abbildung 3.30.:** Ein weiteres Beispiel für die Beweglichkeit innerhalb der Matrix. Dargestellt und markiert wie in Abbildung 3.31, sieht man hier, daß das markierte Sauerstoff im Simulationszeitraum sehr viel Raum „besucht“. Das hat zur Folge, daß Ortskorrelationen mit diesem Sauerstoff extrem klein ausfallen, obwohl keine Diffusion stattfindet. Möglich wird diese hohe Beweglichkeit dadurch, daß sich das Tetraeder mit dem markierten Teilchen sich um die beiden Si-O-Si Bindungen recht gut drehen und diese auch im Winkel beugen kann.



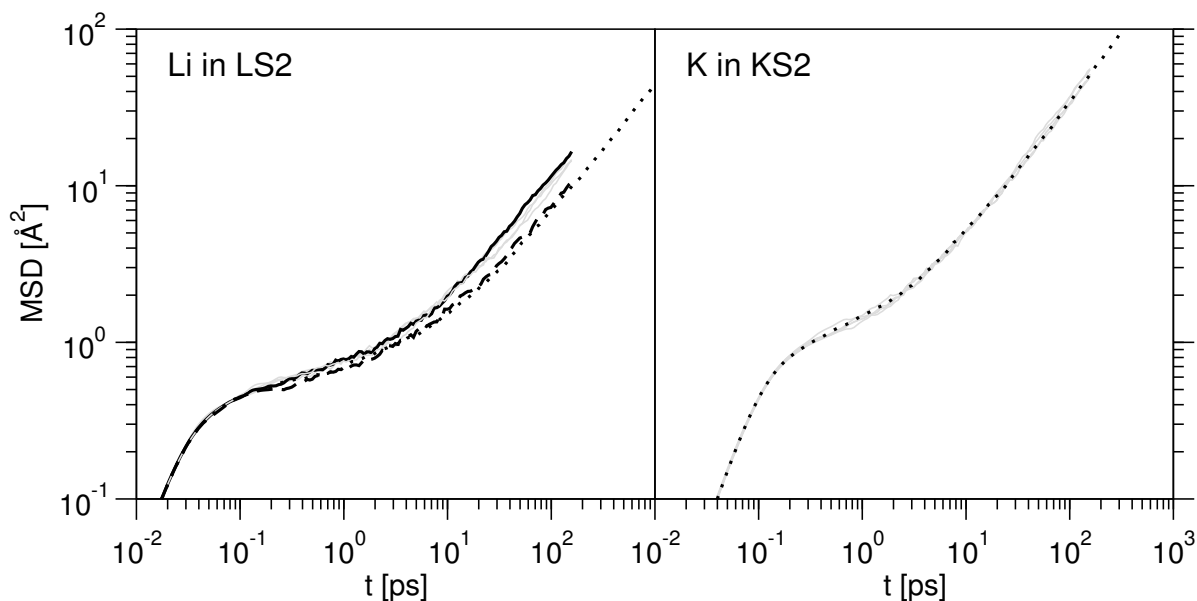


**Abbildung 3.31.:** Links ist das Squared-Displacement eines einzelnen Sauerstoffatoms geplottet, welches schon in Abbildung 3.29 dargestellt wurde (vgl. auch wegen Art der Darstellung). Die gestrichelte Linie gibt das Quadrat des NN-Abstands von O–O wieder. Rechts sieht man einen Schnappschuß der Konfiguration um eben dieses, blau markierte, Sauerstoffatom herum. Hier wurde zur besseren Übersichtlichkeit auf die Alkaliionen verzichtet, da sie das zentrale Tetraeder direkt umgeben. In weiß wurde die Trajektorie des markierten Teilchens eingezeichnet. Anhand der Trajektorie erkennt man ein Schrittmotorartiges Verhalten des Tetreders, welches sich um die untere Si–O–Si Bindung in ca.  $\frac{1}{3}\pi$  dreht.

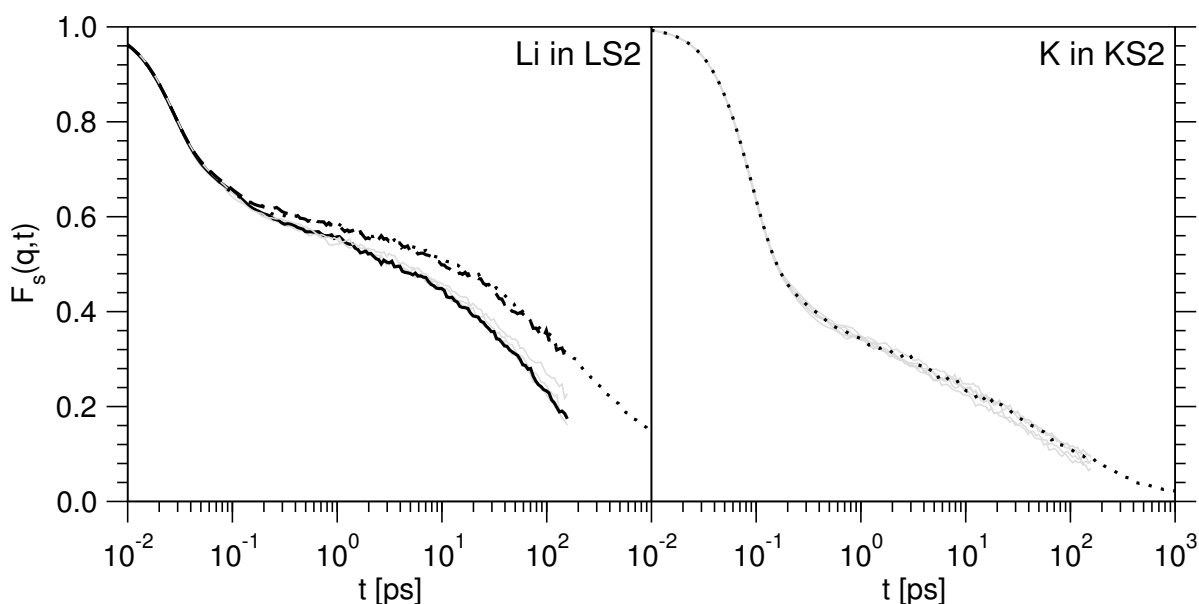
ist das eben erwähnte Tetraeder, etwas vergrößert und gekippt dargestellt, so daß die drei mit Ionen koordinierten O–Atome sichtbar sind (in dieser Darstellung sind die O–Atome am roten Ende der Zylinder angeordnet).

Zur weiteren Untersuchung von Alterungs-Effekten wurden für KS2, LKS2 und LS2 bei 1000 K Äquilibrierungsläufe über ca. 80 ns durchgeführt. Diesem, für MD-Simulationsmaßstäbe, sehr langem Simulationslauf wurden bei 16 ns, 32 ns, 47 ns, 63 ns und 80 ns Konfigurationen entnommen. Diese Konfigurationen dienten als Ausgangspunkt für mikrokanonische Auswertungsläufe über ca. 160 ps. Ziel ist es, zu erkennen, ob und welche Auswirkungen die fortschreitende Äquilibrierung des Systems hat. Dazu wurden auf Basis der äquilibrierten Konfigurationen die MSD's (Abbildung 3.32) und die  $F_s(q, t)$  (Abbildung 3.33) bestimmt. Betrachtet man in beiden Abbildungen jeweils die Resultate für K in KS2, so erkennt man nur statistische Abweichungen der Resultate, d.h. das Verhalten der MSD's und der kohärenten Streufunktionen stellt sich, unabhängig von der Länge der Äquilibrierung, gleich dar. Wendet man sich jetzt aber den Daten von Li in LS2 zu, so erkennt man, daß zwischen 63 ns und 80 ns eine folgenschwerere Änderung im System stattgefunden haben muß. Man erkennt in der Dynamik, die auf der Konfiguration nach 80 ns Äquilibrierung basiert, eine andere Diffusionskonstante ( $D = 9.8 \cdot 10^{-7} \text{ cm}^2/\text{s}$ ) als bei dem Auswerteläufen bis zu 63 ns ( $D = 17.7 \cdot 10^{-7} \text{ cm}^2/\text{s}$ ). Bei den  $F_s(q, t)$  erkennt man ein analoges Verhalten, nämlich, daß der Zerfall der Korrelation für die Simulationen, die auf der Konfiguration nach 80 ns Äquilibrierung basieren, deutlich langsamer ist, als für die Simulation auf Basis kürzer äquilibrierter Konfigurationen. Das bedeutet, daß auch die Relaxationszeiten der Teilchen abhängig von dem Grad der Äquilibrierung im System sind.

Die Beobachtung, daß in KS2 unabhängig von der Äquilibrierungszeit die Dynamik unverändert bleibt, wohingegen in LS2 sich Veränderungen zeigen, darf aber nicht mit einer Abhän-



**Abbildung 3.32.:** Links: MSD's für Lithium in LS2 bei 1000 K. Ausgangskonfigurationen wie im Text ab Seite 78 beschrieben. Die durchgezogene Linie basiert auf einer Konfiguration nach ca. 47 ns Äquilibration. Die gestrichelte und die gepunktete Linie haben dieselbe ca. 80 ns äquilibrierte Konfiguration als Grundlage. Die grauen Linien sind die MSD's von Simulationen auf Basis von Konfigurationen nach 16 ns, 32 ns und 63 ns. Sie stimmen im wesentlichen mit dem MSD auf Basis der Konfiguration nach 47 ns überein. Rechts: MSD's für Kalium auf Basis von Konfigurationen bei 16 ns, 32 ns, 47 ns, 63 ns (grau) und 80 ns (grau und schwarz gepunktet). Ihre Verläufe stimmen alle überein.



**Abbildung 3.33.:** Vergleiche Abbildung 3.32 für die Beschreibung der Zeitpunkte und Systeme, zu denen die Kurven gehören. Statt MSD's wurden hier die  $F_s(q, t)$  aufgetragen.

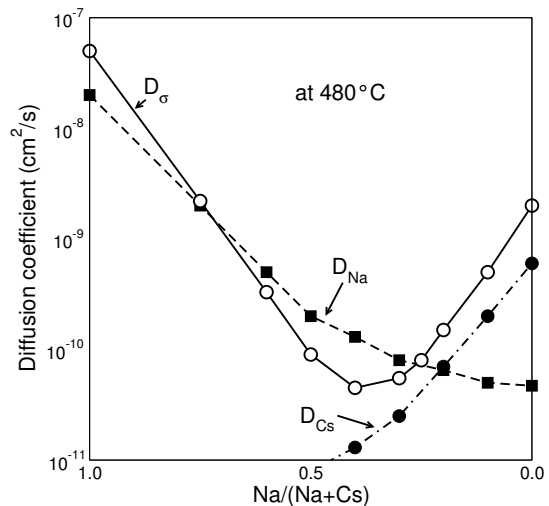
gigkeit vom betrachteten System verwechselt werden. Auch in LS2 sieht man, daß sich bis zu 63 ns keine Veränderungen durch die Äquilibration ergeben haben. Folgerichtig hat man es hier mit Relaxationsprozessen zu tun, die irgendwann während der Äquilibration auftreten. Diese haben zur Folge, daß die Dynamik in Abhängigkeit fortschreitender Äquilibration sich verlangsamt. Durch dieses Verhalten kann man jetzt auch die Diskrepanz im Verhalten der Diffusionskonstanten in der Simulation, die über den ganzen Temperaturbereich ein Arrheniusverhalten zeigen, und den experimentellen Ergebnissen, die, wie in Abbildung 3.21 zu sehen, ein Abknicken zu höheren Aktivierungsenergien zeigen, schlüssig erklären. In dem Temperaturbereich, in dem man im Experiment ein Abknicken beobachtet, fällt in der Simulation die Matrix aus dem Gleichgewicht. Die Folge davon sind zu hohe Diffusionskonstanten, und damit dann auch ein anderes Verhalten der Diffusionskonstanten als im Experiment. Wie man aber sieht, verändern sich auch die Diffusionskonstanten (und Relaxationszeiten) bei längerem Äquilibrieren so, wie man es aus dem Experiment erwartet. Eine systematische Untersuchung der Alterungsphänomene war im Rahmen dieser Arbeit nicht machbar, da dies viele unabhängige Simulationsläufe (in der Größenordnung von 100) erfordern würde, um eine brauchbare Statistik zu erzielen. Für eine derartige Untersuchung ist die Simulation kleinerer Systeme (in der Größenordnung von 1000 Teilchen) vorzuziehen.

Zusammenfassend ergibt sich folgendes Bild: Es handelt sich bei der in der Matrix beobachteten Alterung nicht um diffusive Prozesse. Im wesentlichen handelt es sich hier um Umordnungen von ganzen Matrixbausteingruppen (z.B. ganze  $\text{SiO}_4$ -Tetraeder). Diese geschehen in verschiedenen Ausprägungen. Zum einen gibt es sehr bewegliche Unterelemente in der Matrix, die vor allem den O-Atomen sehr viel Spielraum lassen, ohne die Bindungen brechen. Beispiele dafür wurden in Abbildung 3.30 und Abbildung 3.29 vorgestellt. Zum anderen gibt es tatsächlich Prozesse, in denen Bindungen aufgebrochen werden, damit eine strukturelle Umordnung hin zu einem anderen Teil der Matrix möglich wird. Ein Beispiel dafür ist in Abbildung 3.27 gezeigt. Solche Prozesse sind aber zum einen meist nicht besonders langlebig (vgl. Abbildung 3.28) und zum anderen handelt es sich dabei nicht um einen Diffusionsprozeß einzelner Teilchen, wie man es aus den SQD's schließen könnte, sondern um eine Umordnung größerer Elemente der Matrix (Defekte). Diese entstehen, da im Gegensatz zum Experiment in der Simulation nicht so lange äquilibriert werden kann, bis die Matrix im Gleichgewicht ist. Das hat entsprechend auch Auswirkungen auf die Potentiallandschaft, die die Alkaliionen sehen, weshalb deren  $F(q, t)$  synchron zu den  $F(q, t)$  der Matrixelemente Alterung aufweisen. Und selbst wenn die Defekte wieder ausheilen, soll heißen, wieder in ihre Bindungen, wie sie zur Zeit  $t = 0$  waren, zurückkehren, so haben sich sowohl die Matrix, vor allem aber die Alkaliionen aufgrund ihrer hohen Mobilität, bereits nachrelaxiert. Das hat zur Folge, daß die Anfangskonfiguration der Matrix nie exakt wiederhergestellt wird, was den Verlust an Korrelation in  $F(q, t)$  und anderen Größen erklärt.

Betrachtet man die Abhängigkeit dynamischer Größen wie Diffusionskonstanten und inkohärente Streufunktionen von der Länge der Äquilibration, so stellt man fest, daß es bei langem Äquilibrieren zu Relaxationen kommt, die dafür sorgen, daß die dynamischen Größen sich so verhalten, wie man es aus Experimenten kennt. Das bedeutet, daß z.B. die Diffusionskonstanten bei längerem Äquilibrieren kleiner werden. Damit sieht man, daß die Tatsache, daß die Matrix sich nicht im Gleichgewicht befindet, wie es im Experiment der Fall ist, durchaus weitreichende Konsequenzen für die Interpretation von Simulationsergebnissen zur Dynamik hat.

## 4. Binäres Lithiumdisilikat mit einem Fremdalkalium

**Abbildung 4.1.:** Experimentelle Ergebnisse zum MAE in einem  $\text{Na}_2\text{O} - \text{Cs}_2\text{O} - \text{SiO}_2$  Glas [134].  $D_{\text{Na}}$  und  $D_{\text{Cs}}$  geben die Selbstdiffusionskonstante von Natrium und Cäsium an,  $D_\sigma$  ist die Gesamtdiffusivität, errechnet aus Leitfähigkeitsmessungen.



Nach der Untersuchung rein binärer Alkalidisilikate soll nun ein Extremfall des MAE untersucht werden. Wenn in der Literatur vom MAE die Rede ist, so wird meist auf den dramatischen Abfall der Leitfähigkeit bzw. Gesamtdiffusion bei einer Mischung von Alkaliionen verwiesen. Dort wird aber meist nur auf den Abfall des kollektiven MAE verwiesen, also auf die Gesamtdiffusion aller Alkalikomponenten, aber der MAE für eine der beteiligten Alkaliionen ist am stärksten, wenn der Konzentrationsanteil dieser Komponente gegen 0 geht.

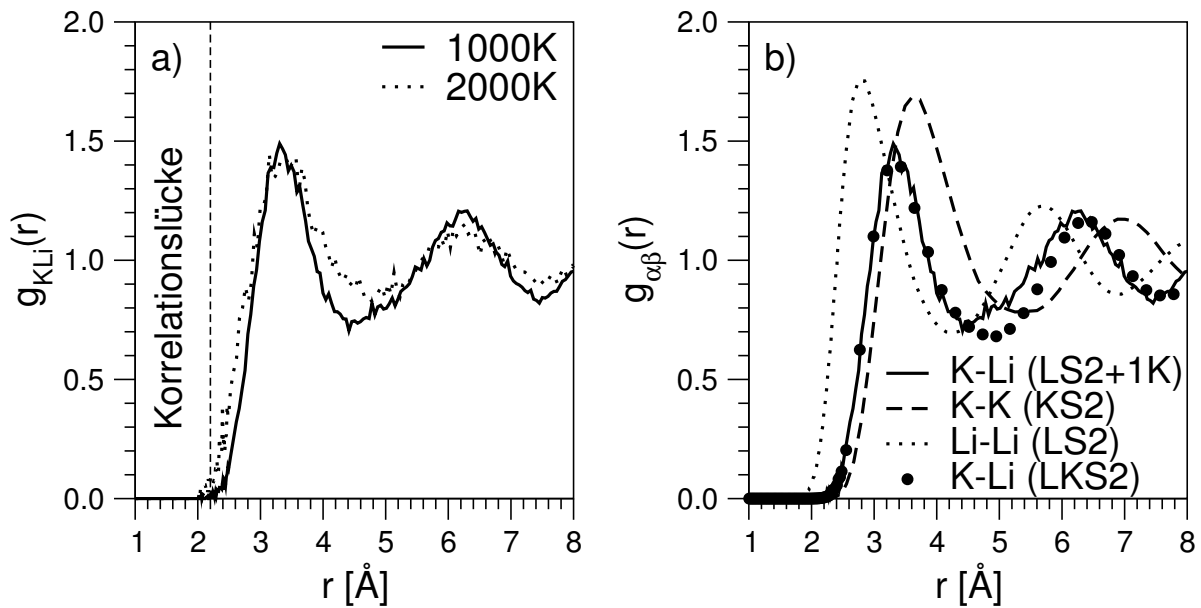
Dies soll anhand der Abbildung 4.1 mit experimentellen Ergebnissen von Messungen an  $\text{Na}_2\text{O} - \text{Cs}_2\text{O} - \text{SiO}_2$  von Terai [134] dargelegt werden. Dort sieht man die aus Leitfähigkeitsmessungen und der Nernst-Einstein-Relation

$$D_\sigma/\sigma = kT/Ne^2,$$

mit  $N$  der Anzahl der Ionen pro Einheitsvolumen und  $e$  der Elektronenladung, resultierende Diffusivität  $D_\sigma$  (durchgezogene Linie) als auch die Einzeldiffusivitäten  $D_{\text{Na}}$  und  $D_{\text{Cs}}$  sowie der Leitfähigkeit  $\sigma$ . Anhand dieser Daten erkennt man, daß der MAE, z.B. für Natrium, am stärksten auftritt, wenn die Konzentration des Natriums niedrig ist. Das bedeutet, im Grenzfall von einem Na-Ion in, z.B., einem Cs-Silikat, ist der MAE für das eine Na-Ion maximal.

Dieses Szenario soll in diesem Kapitel untersucht werden. Dabei wird ein LS2-System untersucht, bei dem ein Li-Atom durch ein K-Atom ersetzt wurde (LS2+1K). Ziel ist es, Besonderheiten des MAE bei Vorhandensein eines fremdnamigen Alkali-Teilchens zu erkennen, die bei der Untersuchung des kollektiven MAE in Kapitel 5 hilfreich sein können.





**Abbildung 4.2.:** a) Über 20 LS2+1K–Systeme gemittelt  $g_{\text{KLi}}(r)$  (bei 1000 K und 2000 K). Gut zu erkennen ist die Korrelationslücke bis ca. 2.2 Å (gestrichelte Linie). b) Vergleich von  $g_{\text{KLi}}(r)$  mit  $g_{\text{KK}}(r)$  in KS2 und  $g_{\text{LiLi}}(r)$  in LS2 (alles bei 1000 K). Als Kreise sind die Daten für  $g_{\text{KLi}}(r)$  in LKS2 aufgetragen.

## 4.1. Strukturelle Besonderheiten um das Fremdion

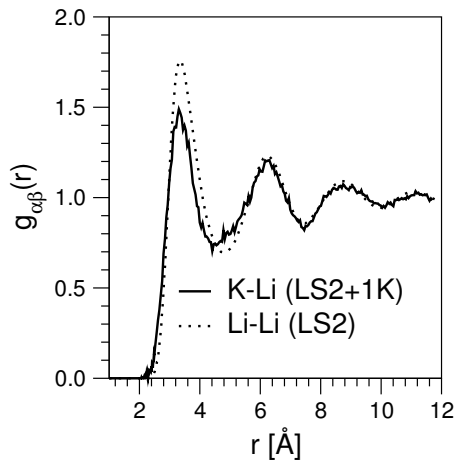
Da wir im wesentlichen ein LS2–System mit einer Störung, dem Kaliumion, vorliegen haben, konzentriert sich die folgende Betrachtung der Struktur auf die Umgebung des Kaliumions.

Gerade für die Betrachtung der lokalen Struktur um das Kalium eignen sich die Paarkorrelationsfunktionen  $g_{\alpha\beta}(r)$ .<sup>1</sup> In Abbildung 4.2.a) sind die K–Li–Korrelationen für 1000 K und 2000 K aufgetragen. Dort ist die Korrelationslücke bis ca. 2.2 Å zu erkennen. Wie zu erwarten, verbreitert sich die Verteilung hin zu höheren Temperaturen und die Maxima und Minima nähern sich  $g_{\alpha\beta}(r) = 1$ .<sup>2</sup> In Abbildung 4.2.b) wird  $g_{\text{KLi}}(r)$  mit  $g_{\text{LiLi}}(r)$  aus LS2,  $g_{\text{KK}}(r)$  aus KS2 und  $g_{\text{KLi}}(r)$  aus LKS2 verglichen. Das erste NN–Maximum stellt sich dabei zwischen den ersten NN–Maxima von  $g_{\text{LiLi}}(r)$  und  $g_{\text{KK}}(r)$  ein, im Vergleich zu  $g_{\text{KLi}}(r)$  aus LKS2 ist das erste NN–Maximum in Form und Höhe identisch. Also hat die Tatsache, daß keine weiteren Kaliumionen im System vorhanden sind, keinen Einfluß auf die lokale Nachbarschaft mit Lithium.

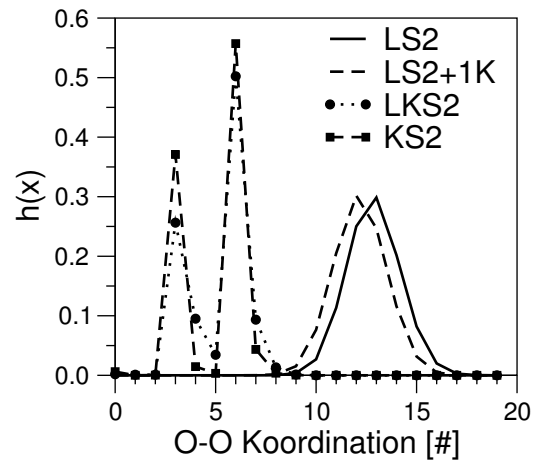
Das erste NN–Maximum von  $g_{\text{KLi}}(r)$  liegt hier bei 3.34 Å. Da nur ein Kaliumatom vorhanden ist und sich somit alle weiteren NN–Schalen rein mit Lithium auffüllen, ist zu erwarten, daß alle Peaks und Minima nach dem ersten NN–Peak, dem von  $g_{\text{LiLi}}(r)$  in LS2 entsprechen. Diese Vermutung trifft zu, wie man in Abbildung 4.3 sehen kann. Dabei wurde  $g_{\text{LiLi}}(r)$  um 0.55 Å zu höheren  $r$  verschoben. Man erkennt ab dem 2. NN–Maximum eine fast perfekte Übereinstimmung von  $g_{\text{LiLi}}(r)$  in LS2 und  $g_{\text{KLi}}(r)$  in LS2+1K.

<sup>1</sup>vgl. dazu Kapitel 2.2.1.1.

<sup>2</sup>Es sei daran erinnert, daß im Fall des idealen Gases überall  $g(r) = 1$  gilt.



**Abbildung 4.3.:** Superposition von  $g_{KLi}(r)$  in LS2+1K und  $g_{LiLi}(r)$  in LS2 durch Verschiebung von  $g_{LiLi}(r)$  um  $0.55 \text{ \AA}$  zu höheren  $r$ .



**Abbildung 4.4.:** O–O–Koordination in LS2, LS2+1K, LKS2 und KS2.

Daraus kann man folgern, daß das Kaliumion keine weitreichenderen strukturellen Veränderungen induziert als bis in die 2. NN–Schale von  $g_{KLi}(r)$ . Bis auf eine Veränderung der Abstände in unmittelbarer Umgebung um das Kalium, schlicht durch die andere Wechselwirkung zwischen K und Li als zwischen Li und Li ausgelöst, verläuft  $g_{KLi}(r)$  ab ca.  $5.5 \text{ \AA}$  genau so, wie  $g_{LiLi}(r)$  in LS2. Auch ein Vergleich aller anderen partiellen Paarkorrelationen zwischen LS2 und LS2+1K ergaben keine Veränderungen in der Struktur.

Doch so lokal der Einfluß des Kaliumions bei der Betrachtung von  $g_{\alpha\beta}(r)$  auch erscheint, treten in dem System weitreichendere Veränderungen auf, als man es zunächst vermuten würde.

Einen sehr interessanten Effekt kann man nämlich bei der Betrachtung der Koordinationszahlverteilungen<sup>3</sup> beobachten. Denn die Koordination von O–O ändert sich vom binären System zum System mit nur einem Fremdkalium deutlich (Abbildung 4.4). In LS2+1K verschiebt sich der Peak der O–O–Koordination von 13 auf 12 NN und die Verteilung generell hin zu niedrigeren Koordinationen.<sup>4</sup>

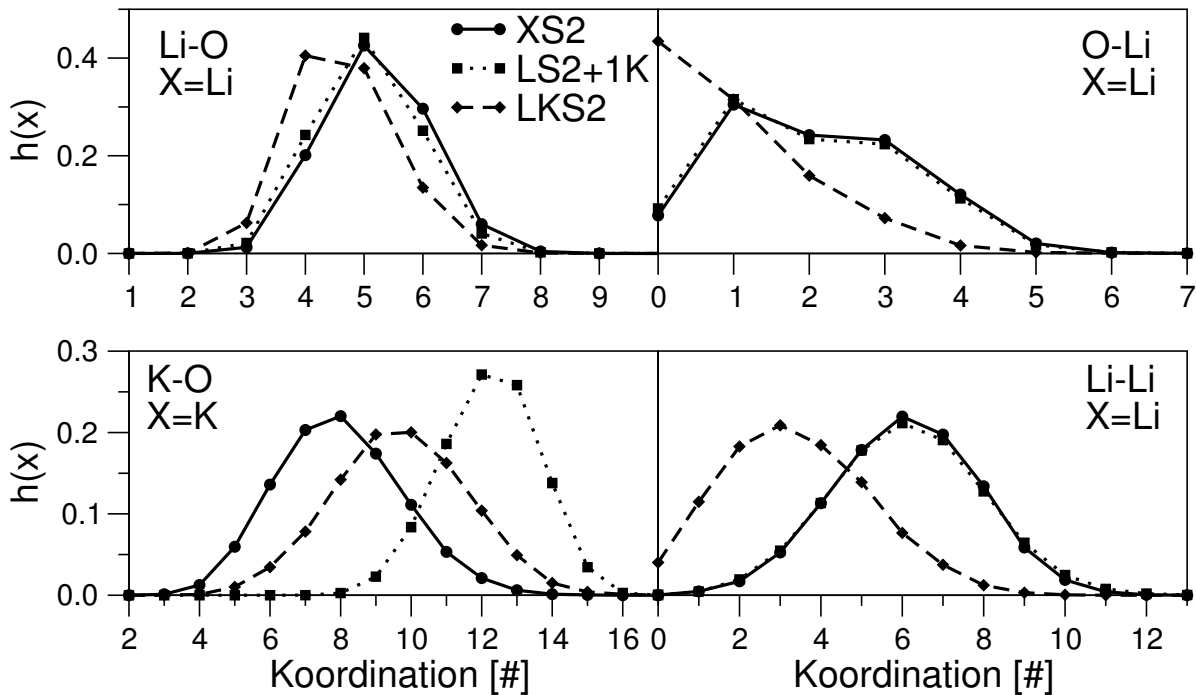
Die Dichte kommt als Grund für die geringere Koordination nicht in Frage. Im Mittel ist sie für die kleinen Systeme, die hier gerechnet wurden (1152 Teilchen), etwas höher ( $\rho = 2.46 \text{ g/cm}^3$ ) als im reinen LS2–System mit 8064 Teilchen ( $\rho = 2.43 \text{ g/cm}^3$ ). Doch die höhere Dichte läßt eigentlich eine höhere Koordination erwarten. Der Vergleich der Paarkorrelation  $g_{OO}(r)$  zwischen LS2 und LS2+1K ergeben auch keinen Hinweis, da diese praktisch identisch sind.

Die Frage ist, was nun der Auslöser dieser Veränderung ist. Es sind zunächst nur zwei „technische“ Unterschiede zwischen LS2 und LS2+1K zu nennen: Einmal erhält man durch die Definition des Potentials leicht verschiedene Sauerstoffladungen<sup>5</sup>, zum anderen wurde ein

<sup>3</sup>vgl. Kapitel 2.2.1.2.

<sup>4</sup>Einen Überblick über die Veränderung anderer Koordinationen bietet Abbildung 4.5.

<sup>5</sup>vgl. Kapitel 2.1.2.1.



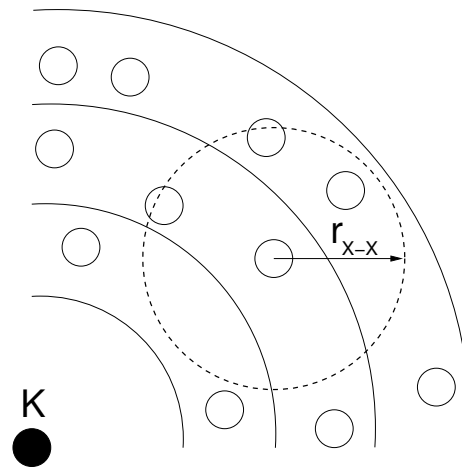
**Abbildung 4.5.:** Verschiedene Koordinationen bei 1000 K. Die vorgestellten Li-Koordination tendieren bei Zugabe von Kalium hin zu kleineren Koordinationszahlen, wohingegen die K-O-Koordination eine stark steigende Koordination mit steigender Li-Konzentration aufweist.

Lithium- durch ein Kaliumion ersetzt, welches eine andere Koordination und Wechselwirkung mit Sauerstoff aufweist, weshalb die mittlere Koordination im System sich verändert.

Zunächst einmal zum Ladungsunterschied des Sauerstoffs nach der Formel die in Tabelle 2.4 angegeben ist: Die LS2+1K-Systeme enthalten gesamt jeweils 1152 Teilchen (256 Si, 1 K, 255 Li, 640 O). Das bedeutet, daß im reinen LS2 die Sauerstoffladung  $q_O = 1.308$  ist, während man in LS2+1K den Wert  $q_O = 1.30797$  hat. Man sieht, der Unterschied ist so klein, daß der Ladungsparameterunterschied keine sichtbare Rolle spielen sollte. Bleibt noch der Unterschied, der durch den Tausch eines Lithiumions durch ein Kaliumion entstanden ist. Dies ist schlüssig, da in Abbildung 4.4 zu sehen ist, daß für einen steigenden Kaliumanteil (vor allem in LKS2 und KS2) die O-O-Koordination zu deutlich kleineren Koordinationszahlen und sogar von einer Gaußartigen Verteilung zu einer Doppelpeakstruktur übergeht. Damit passt die niedrigere O-O-Koordination gut ins Bild, was aber erstaunlich ist, ist der große Effekt, der durch die Ersetzung eines Teilchens in ein LS2+1K entstanden ist. Die mittlere O-O-Koordination ändert sich ja von 12.87 in LS2 zu 12.19 in LS2+1K. Das ist eine Änderung fast in der Größenordnung von 1 in der mittleren Koordination, ausgelöst durch einen Ionentausch in der Größenordnung von  $\approx 4\%$ .

Von großem Interesse ist jetzt natürlich die Frage, wie weit der Einfluß des Fremdions (Kalium) auf die Koordination reicht.

Dazu wurde eine Auswertung gemacht, die anhand von Abbildung 4.6 erläutert werden soll. Die Frage lautet, wie weit der Einfluß des Kaliumions (schwarz) reicht, um Koordinationen von Teilchen der Sorte X (weiße Kreise) zu beeinflussen. Mittels der Daten aus  $g_{KX}(r)$  er-



**Abbildung 4.6.:** Skizze zur Veranschaulichung der NN-Suche in Abhängigkeit vom Abstand zum Fremdkaliumion (vgl. Seite 84).

mittelt man die Minima der NN-Abstände, um die X-Teilchen in die ersten drei K-X-NN-Schalen zu unterteilen.<sup>6</sup> Dann wird über  $g_{XX}(r)$  der X-X-NN-Abstand  $r_{XX}$  ermittelt. Jetzt kann man schalenabhängig die Koordinationszahlen für X-X berechnen, also wieviele gleichnamige NN ein X-Teilchen im Abstand der  $n$ -ten NN-Schale vom Kaliumion hat.

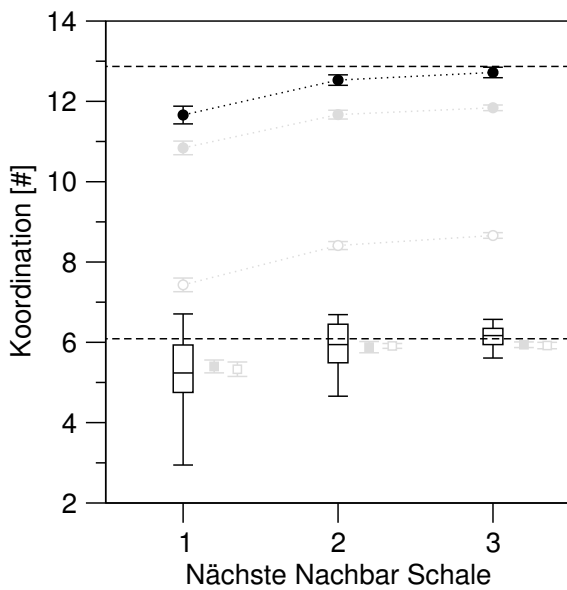
Diese Auswertung wurde für X=Li und X=O in Abbildung 4.7 festgehalten. Betrachtet werden zunächst die Ergebnisse für die O-O-Koordination (Kreise). Wie zu erwarten, ist der Einfluß des Kaliumions auf die Koordination in der 1. NN-Schale am stärksten, hört dort aber nicht auf. Auch in der 2. NN-Schale läßt sich noch eine leicht geringere Koordination feststellen, d.h. die Anwesenheit des Kaliumions hat noch Auswirkungen auf die chemische Umgebung von Sauerstoffen im Abstand von ca. 6– bis 8.5 Å. Interessanterweise nimmt der Einfluß des Kaliumions auch bei steigender Temperatur absolut kaum ab, im Gegenteil, rechnet man die relative Änderung der Koordinationszahl aus, so steigt der Einfluß des Kaliumions auf die O-O-Koordination mit steigender Temperatur.<sup>7</sup>

Aufgrund der großen unsymmetrischen Streuung der Ergebnisse der Li-Li-Koordination bei 1000 K werden die hierfür ermittelten Ergebnisse als *Boxplots* dargestellt. Das ist nötig, weil man es hier nicht mit der mehrfachen Messung einer Observablen zu tun hat, bei der ein guter Mittelwert mit möglichst kleinen (Meß-)Fehlern ermittelt werden soll, sondern man hat es mit 20 Einzelereignissen zu tun. Eine Beschreibung des Aufbaus von Boxplots findet sich im Infokasten auf Seite 86.

Man erkennt so auch, daß offensichtlich sehr häufig die Li-Li-Koordinationszahl in den ersten zwei Schalen um das Kaliumion von diesem beeinflusst wird. Dasselbe sieht man dann auch für die höheren Temperaturen (2000 K und 4000 K, geschlossene und offene Quadrate). Bei der 1. NN-Schale kann man argumentieren, daß die geringere Li-Li-Koordination durch das Ersetzen eines Lithiumions durch ein fremdnamiges Kaliumion zustande kommt, weshalb einfach in der Zählung eins fehlt, um den Mittelwert der NN in LS2 zu erreichen. Diesen

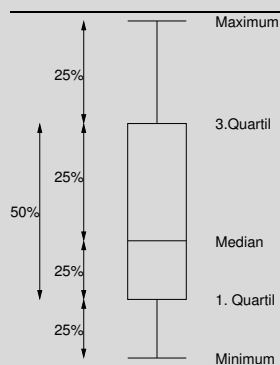
<sup>6</sup>vgl. Kapitel 2.2.1.1.

<sup>7</sup>In diesem Zusammenhang darf man allerdings nicht vergessen, daß sich die Dichte auch mit der Temperatur verändert. Vgl. hierzu Abbildung 3.1.



**Abbildung 4.7.:** Ergebnisse zur NN-Suche, wie sie auf Seite 84 und in Abbildung 4.6 beschrieben wurde (bei 1000 K). Die schwarzen Punkte mit den Fehlerbalken geben hier die O–O-Koordinationszahl an (grau und gefüllt: 2000 K; graue Kreise: 4000 K). Die Boxplots repräsentieren die Li–Li-Koordinationszahlen. Diese Darstellung wurde in diesem Fall deshalb gewählt, weil eine Standardabweichung nicht die Asymmetrie der Verteilung, speziell hin zu niedrigen Koordinationen, wiedergegeben hätte. Die gestrichelten Linien sind die Mittelwerte der Koordinationen, die in reinem LS2 bei 1000 K ermittelt wurden. Die Li–Li-Koordinationszahlen für 2000 K und 4000 K wurden aufgrund ihrer kleinen Streuung mittels grau und weiß gefüllter Quadrate mit Fehlerbalken dargestellt.

### Der Boxplot.



Der Boxplot, auch Schachteldiagramm genannt, ist eine graphische Darstellung metrischer Daten. Am Boxplot kann man sowohl *Median*, oberes und unteres *Quartil* als auch die *Extremwerte* der Daten ablesen. Durch den Aufbau und die im Boxplot enthaltenen Quartile ist es möglich, einen ersten Eindruck von Lage und Form der Verteilung zu erlangen. Im speziellen eignet sich der Boxplot zum Lagevergleich mit anderen Boxplots.

**Median:** Sortiert man eine Reihe von Meßwerten der Größe nach, so ist im Falle einer ungeradzahligen Anzahl von Meßwerten der Wert, der in der Mitte der Reihe liegt, der Median. Hat man eine geradzahlige Menge von Meßwerten, so ist der Median das arithmetische Mittel der mittleren Werte. Diese Vorgehensweise machen den Median zu einem relativ Ausreisserunempfindlichen Maß für das Zentrum der Verteilung.

**Quartil:** Sortiert man eine Reihe von Meßwerten der Größe nach, und ist die Länge  $2n$ , so ist der Median der  $n$  kleinsten Werte das 1. Quartil. das 2. Quartil ist der Median der  $2n$  Werte und das 3. Quartil unterteilt die  $n$  größten Werte analog zum 1. Quartil. Beträgt die Menge der Meßwerte  $2n + 1$ , so verwendet man zur Bestimmung des 1. und 3. Quartils jeweils die  $n + 1$  kleinsten und größten Werte. Die Quartile unterteilen eine Verteilung in Viertelanteile.

Einwand kann man auch auf die Koordinationen bei 2000 K und 4000 K aufgrund ihrer geringen Schwankung und der in der 2. Schale schon fast am LS2-Wert liegenden Koordination gelten lassen. Bei 1000 K zeigen die Boxplots jedoch, daß in ca. 25 % der untersuchten Fälle der Unterschied deutlich größer ist, als die Argumentation des einen in der Zählung fehlenden Lithiumatom zulassen würde.

Abschließend kann man bezüglich der strukturellen Änderungen, die das eine Kaliumion in LS2 ausgelöst hat, folgende Ergebnisse festhalten: Zum einen scheint es zunächst, daß das Kaliumion nur einen lokalen Einfluß ausübt, da z.B. in  $g_{\text{KLi}}(r)$  nur in der ersten NN-Schale Auswirkungen zu sehen sind, der Anteil der Paarkorrelation für größere  $r$  jedoch exakt mit dem für  $g_{\text{LiLi}}(r)$  übereinstimmt. Zudem zeigen andere Paarkorrelationen, z.B.  $g_{\text{OO}}(r)$ , gar keine Veränderung im Vergleich zu reinem LS2.

Dem Schluß, daß der Einfluß des fremdnamigen Ions nur lokal ist, steht allerdings die globale Änderung der O–O-Koordination entgegen, welche, im Vergleich zum Anzahlverhältnis von Lithiumionen und dem Kaliumion, einen überproportional großen Einfluß des Kaliumions zeigt.

Um den Bereich um das Kaliumion besser zu untersuchen, wurde eine orts aufgelöste Koordinationsverteilung eingeführt, um Koordinationen in Abhängigkeit von der Entfernung vom Kaliumion zu bestimmen. Speziell die O–O-Koordination zeigt noch bis zu intermediären Längenskalen, d.h. bis zu ca. 8.5 Å, den Einfluß des Kaliumions. Da aber die Sauerstoffkoordination und vor allem die Anzahl der für Bindungen zur Verfügung stehenden Sauerstoffe auch für Lithium wichtig ist, ist davon auszugehen, daß auch die Lithiumionen mindestens bis zu 8.5 Å auch den Einfluß der Kaliumionen „spüren“, so wie man es auch andeutungsweise aus der orts aufgelösten Koordination für Li–Li sieht.

## 4.2. Dynamik des Fremdions

Die Mittel, die zur Untersuchung der dynamischen Eigenschaften des Systems LS2+1K zur Verfügung stehen, stellen sich zunächst etwas eingeschränkter dar, als es im Kapitel für binäre Systeme der Fall war. Für die Matrix und die Lithiumionen stellt sich die Situation weitgehend identisch zu LS2 dar. Aber Platz–Platz–Korrelationen, wie sie z.B. durch  $F(q, t)$  gemessen werden, lassen sich mit Kalium nicht bewerkstelligen, da es nur ein Ion im System gibt. Das bedeutet, für diesen Fall ist  $F(q, t) = F_s(q, t)$ . Im Falle von  $F_s(q, t)$  kann man aber nur noch feststellen, ob das Kaliumion gesprungen ist, d.h. daß der Wert von  $F_s(q, t)$  auf 0 zerfallen ist, also keine Korrelation mit der Position des Teilchens zur Zeit  $t = 0$  mehr besteht, oder es stellt sich ein Plateau ein, anhand dessen man die Lokalisierung des Teilchens an seiner Position messen kann. Diese Informationen erhält man jedoch auch schon durch die MSD's.

Die Berechnung der Lokalisierungslängen  $r_s$  aus den  $F_s(q, t)$  wurde bereits in Kapitel 3.2 vorgestellt. Aus den Plateauhöhen  $h_{\text{Plateau}}$  der MSD's erhält man die Lokalisierungslänge über die Beziehung

$$r_s = \sqrt{\frac{1}{6} h_{\text{Plateau}}}.$$

Die MSD's sollen nun für die 20 verschieden angesetzten Systeme, die bei 1000 K simuliert wurden, in Abbildung 4.8 betrachtet werden.

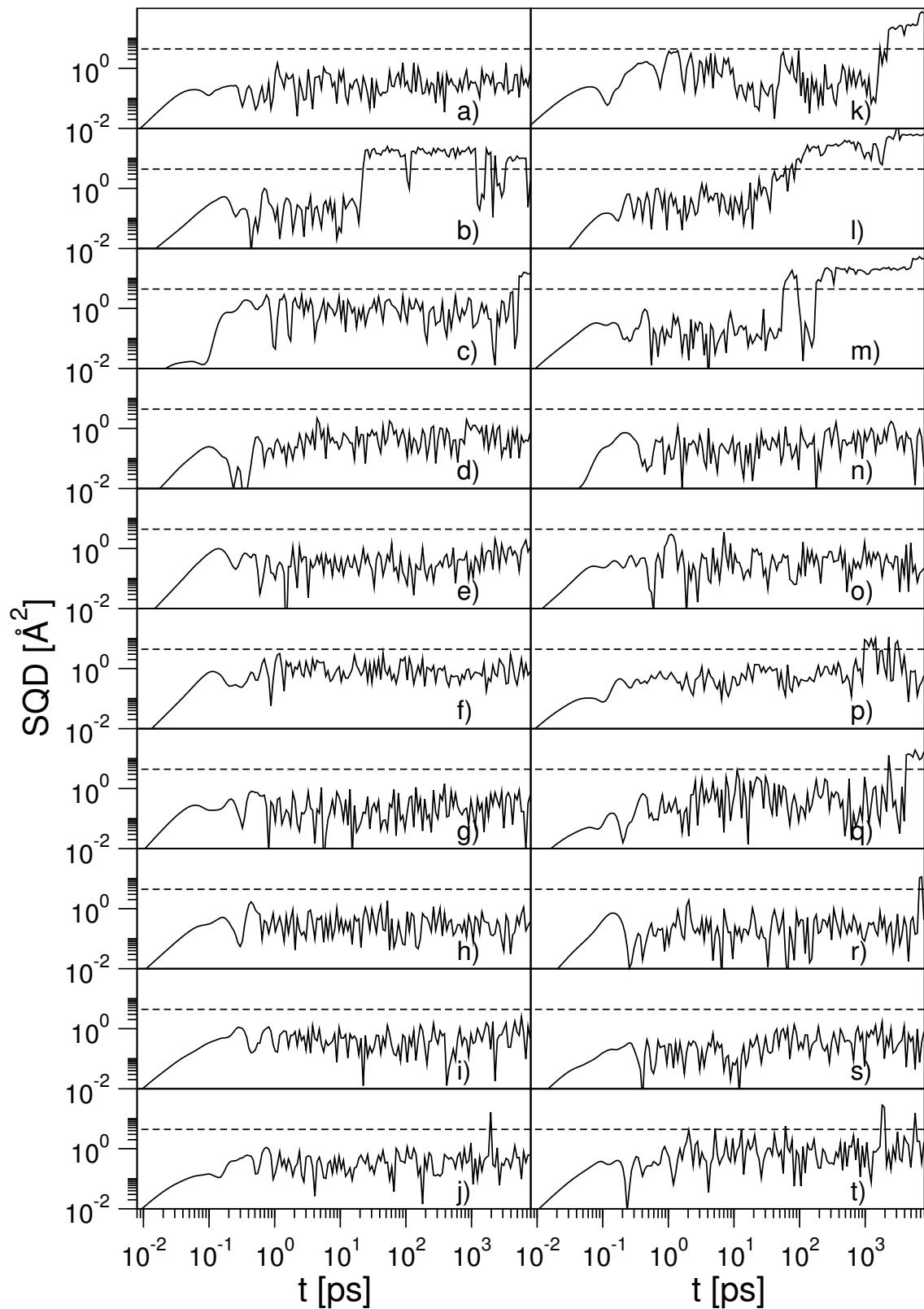
Um Sprünge eines Kaliumions zu identifizieren, wurde folgendes Kriterium verwendet: Die Korrelationslücke der Paarkorrelation  $g_{\text{KLi}}(r)$  beträgt ca.  $2.2 \text{ \AA}$ , bewegt sich also das Kaliumion um diese Distanz von seinem Ort zur Zeit  $t = 0$ , so befindet es sich in der 1. NN–Schale der Li–Ionen und nimmt dort die Position eines Lithiumions ein.<sup>8</sup> Ist also das MSD größer als  $4.4 \text{ \AA}^2$ , so wird das als Sprung gewertet. Ein Beispiel für ein System mit Sprungereignis und für ein System ohne Sprungereignis findet sich in Abbildung 4.9.

Es stellt sich nun die Frage, ob es dynamische Unterschiede zwischen den Systemen mit und ohne Sprung gibt, anhand derer man ein Sprungereignis vorhersagen kann. Die zwei hier relevanten Informationen, die man aus den MSD's direkt ablesen kann, sind, ob ein Sprung stattgefunden hat und wie hoch das Plateau (vor dem Sprungereignis) ist, also wie lokalisiert das Kaliumion ist.

Diese Lokalisierungslängen wurden in Abbildung 4.10 aufgetragen. Anhand der Verteilung der Lokalisierungslängen von Kaliumionen, die springen werden und die nicht innerhalb der Simulationszeit gesprungen sind, kann man keinerlei Vorhersagen machen, da hier keine Präferenzen zu erkennen sind.

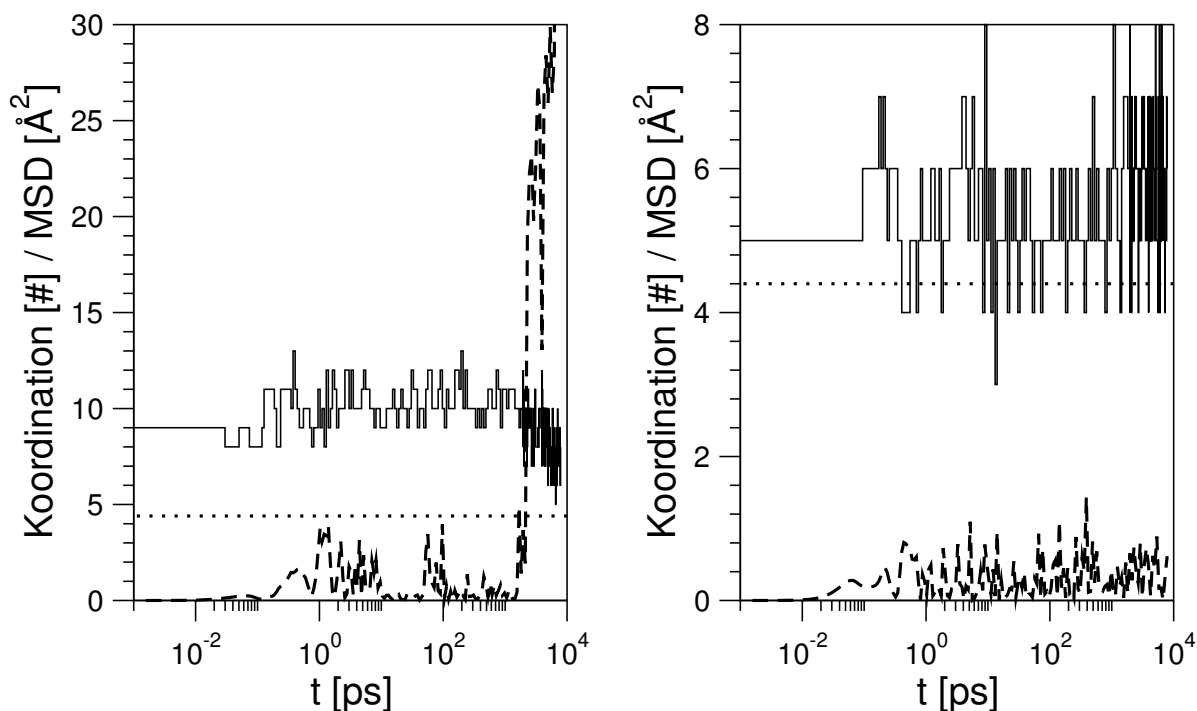
Eine weitere vorstellbare Möglichkeit, warum einige Kaliumionen springen, andere wiederum nicht, ist das Vorhandensein verschiedener chemischer Umgebungen um das Ion. Um das zu überprüfen, wurden die Koordinationszahlen zeitabhängig aufgenommen. Beispiele dafür werden in Abbildung 4.11 vorgestellt. Dort sind die Koordinationszahlabhängigkeiten von Kalium mit Lithium und Sauerstoff aufgetragen. Getrennt sind die Auftragungen nach dem Kriterium, ob im Simulationsverlauf ein Sprungereignis eingetreten ist oder nicht. In schwarz

<sup>8</sup>Dies wurde anhand von  $G_d^{\text{KLi}}(t)$  überprüft.



**Abbildung 4.8.:** Übersicht über alle SQD von Kalium der 20 untersuchten LS2+1K Systeme . Die gestrichelte Linie liegt bei  $4.4 \text{ \AA}^2$ . Sie ist ein Maß dafür, ob ein Teilchen von seinem Platz gesprungen ist. Solche Sprungereignisse beobachtet man in den Systemen b), c), j), k), l), m), p), q), r) und t). Gut sichtbar auch Oszillationen zwischen zwei Plätzen bei b) und Rücksprünge auf die alte Position bei z.B. j).





**Abbildung 4.9.:** Zwei exemplarische Beispiele, anhand derer Sprungereignisse in LS2+1K diskutiert werden sollen (vgl. Text ab Seite 88). Die durchgezogenen Linien repräsentieren die Koordinationszahl von Lithium um das Kaliumion zur Zeit  $t$ . Die gestrichelte Linie ist das MSD des Kaliumions und die gepunktete Linie gibt den Abstand in  $[\text{\AA}^2]$ , der mindestens zurückgelegt werden muß, um auf eine neue Position zu springen. Speziell im Fall mit Sprungereignis (links) kann man nicht wirklich eine signifikante Änderung der Koordination feststellen.

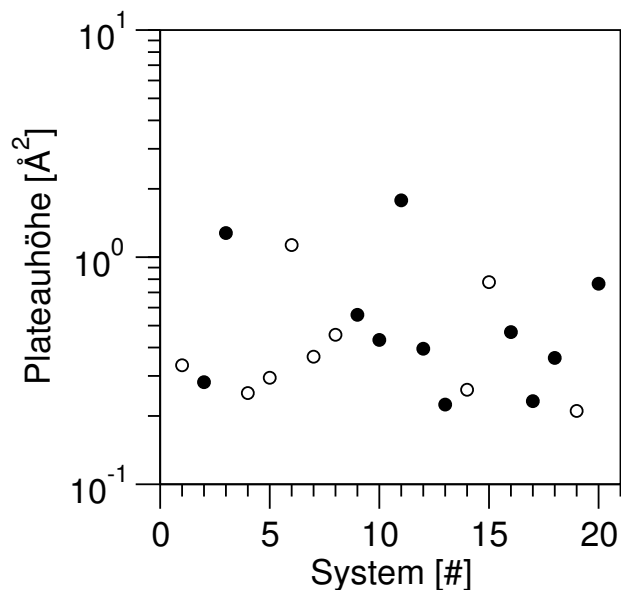
wurde der zeitabhängige Mittelwert, in grau, dunkelgrün und blau die Einzelkoordinations-trajektorien aufgetragen.<sup>9</sup> Wegen der großen Streuung der Trajektorien ist dies sinnvoller als Fehlerbalken.

Es liegen insgesamt hier nur wenige Einzelereignisse (Größenordnung 10 für „mit Sprung“ und „ohne Sprung“) vor, die auch eine große Streuung aufweisen. Daher kann man nur eine vorsichtige Analyse der Ergebnisse aus Abbildung 4.11 machen, die durch größere Statistik zu erhärten oder zu widerlegen wäre: Generell scheint die Streuung der Einzelkoordinations-trajektorien in Systemen ohne Sprünge des Kaliumions kleiner zu sein, als in Systemen mit Sprüngen.

Scheinbar ist im Mittel in Systemen mit Sprungereignissen die mittlere K–Li–Koordination etwas höher und die K–O–Koordination etwas niedriger als in Systemen ohne Sprungereignis. Diese Folgerung ist aber mit Vorsicht zu genießen, da einerseits die Streuung sehr hoch ist und andererseits die Grundgesamtheit der Ereignisse klein, weshalb stochastische Aussagen hier nicht sehr fundiert sind.

Da die Streuung der Koordinationszahlen in Systemen, in denen Sprünge auftreten, deutlich

<sup>9</sup>Warum in den unteren Abbildungen statt grau zwei verschiedene Farben für die Trajektorien benutzt wurden, wird später im Text an entsprechender Stelle erläutert.



**Abbildung 4.10.:** Plateauhöhen der SQD's aus Abbildung 4.8 (vor dem Sprung). Die Plateauhöhen sind ein Maß der Lokalisierung der Kaliumionen. Die offenen Kreise beschreiben die Plateauhöhen der Systeme ohne Sprünge, die ausgefüllten Kreise repräsentieren Systeme mit Sprüngen. Anhand dieser Daten lässt sich kein Rückschluss auf einen direkten Zusammenhang zwischen Lokalisierungslänge und Sprungwahrscheinlichkeit ziehen.

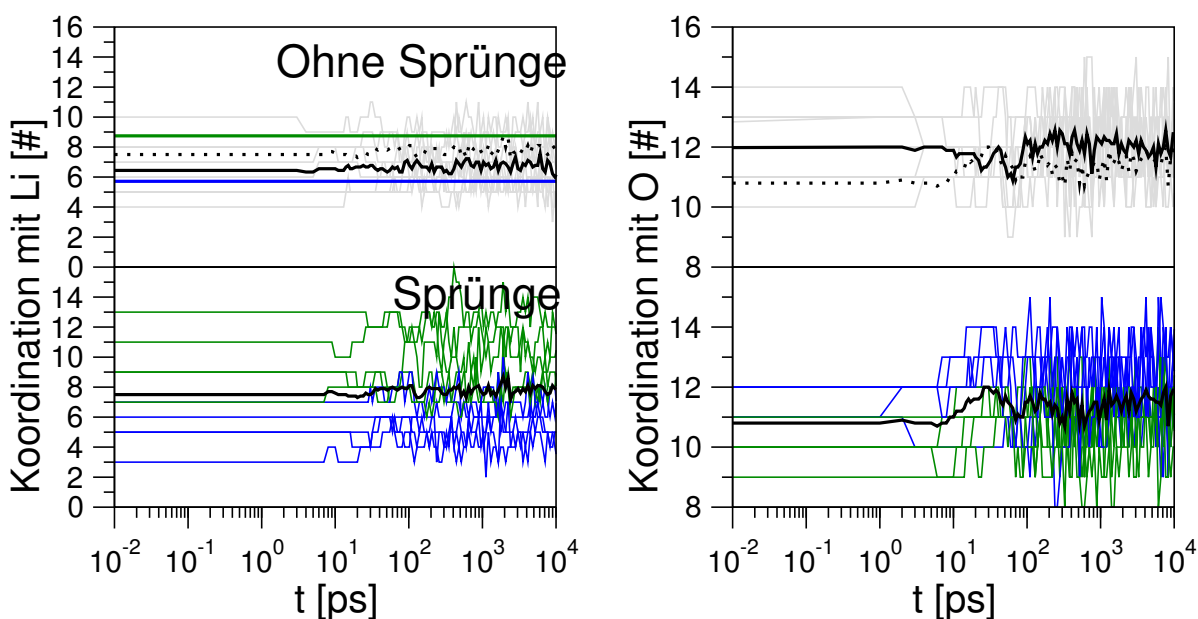
größer zu sein scheint, als in Systemen ohne Sprungereignis, wurden die Koordinationszahltrajektorien genauer betrachtet. Dazu wurden die Trajektorien für die K–Li–Koordination mit Sprungereignissen in zwei Klassen aufgeteilt: Einmal in Trajektorien, die generell oberhalb des zeitlich aufgelösten Mittelwerts<sup>10</sup> liegen (dunkelgrün) und zum anderen in Trajektorien, die generell unterhalb liegen (blau).

Wie man sieht, kann man diese Aufteilung problemlos machen, da die Trajektorien nicht weit und nicht dauerhaft auf die andere Seite der Mittelwerttrajektorie wechseln. Nimmt man jeweils den Mittelwert dieser beiden Klassen von Trajektorien und zeichnet diesen in die Trajektorienplots des Systems ohne Sprünge ein, so sieht man, daß diese Mittelwerte die Koordinationszahltrajektorien großteils einrahmen. Dies gibt Anlaß zur Vermutung, daß ein Sprungereignis dadurch begünstigt wird, daß entweder besonders viele oder besonders wenige erste Lithiumnachbarn um das Kaliumion vorhanden sind.

Erhärtet wird diese Vermutung noch durch folgende Beobachtung: Färbt man die Trajektorien der O–O–Korrelation mit Sprüngen auch ein, und zwar nach dem Kriterium, daß die K–Li–Koordination in eine der vorher erwähnten Klassen sortiert wurde, so ergibt sich, daß, wenn das Kaliumion viele Lithiumnachbarn hat, seine Nachbarschaft mit Sauerstoff generell gering ist. Ist Kalium allerdings von wenig Lithium umgeben, so hat es viel Sauerstoffnachbarn. Die Sprungbereitschaft des Kaliums scheint also wirklich mit seiner direkten chemischen Umgebung korreliert zu sein.

Allerdings muß man mit diesem Ergebnis sehr vorsichtig umgehen. Die Anzahl der Trajek-

<sup>10</sup>weiterhin auch Mittelwerttrajektorie genannt.



**Abbildung 4.11.:** Koordinationszahlen in Abhängigkeit von  $t$ . Separat aufgetragen für Systeme, in denen das Kaliumion gesprungen ist und Systeme in denen kein Sprungereignis auftrat. Links die Koordination K–Li, rechts K–O. Schwarze Linien stellen Mittelwerte der grauen, beziehungsweise dunkelgrünen und blauen, Koordinationstrajektorien dar, die gepunktete Linie oben ist der Mittelwert von der K–Li– bzw. K–O–Koordination im System mit Sprüngen. Die dunkelgrüne und blaue durchgezogenen Linien links oben sind die Mittelwerte der gleichfarbigen Trajektorien unten links.

torien, die in jeweils einer der oben genannten Klassen eingeordnet wurde, beträgt jeweils 5. Damit sind die Ergebnisse hier weit von einer statistischen Gewissheit entfernt und können hier nur als Anregung für genauere Untersuchungen dienen.

Anhand der Ergebnisse dieses Unterkapitels sieht man also, daß die Sprungwahrscheinlichkeit nicht mit der Lokalisierung korreliert zu sein scheint. Aber es gibt Hinweise, daß die chemische Umgebung, namentlich eine hohe bzw. niedrige Anzahl nächster Lithiumnachbarn, gekoppelt mit einer niedrigen bzw. hohen Anzahl an nächsten Sauerstoffnachbarn, ein Sprungereignis begünstigt.

# 5. Ternäres Alkalidisilikat und der MAE

In diesem Kapitel soll es um die Untersuchung des ternären Alkalidisilikats  $(0.5 \cdot \text{Li}_2\text{O})(0.5 \cdot \text{K}_2\text{O})(2 \cdot \text{SiO}_2)$  (LKS2) gehen. Darin ist die molare Gesamtkonzentration der Alkaliionen gleich wie bei den vorhergehenden binären Alkalidisilikaten LS2 und KS2 (33.3 mol%), ist aber halb–halb auf die vorher untersuchten Alkaliionensorten unterteilt.

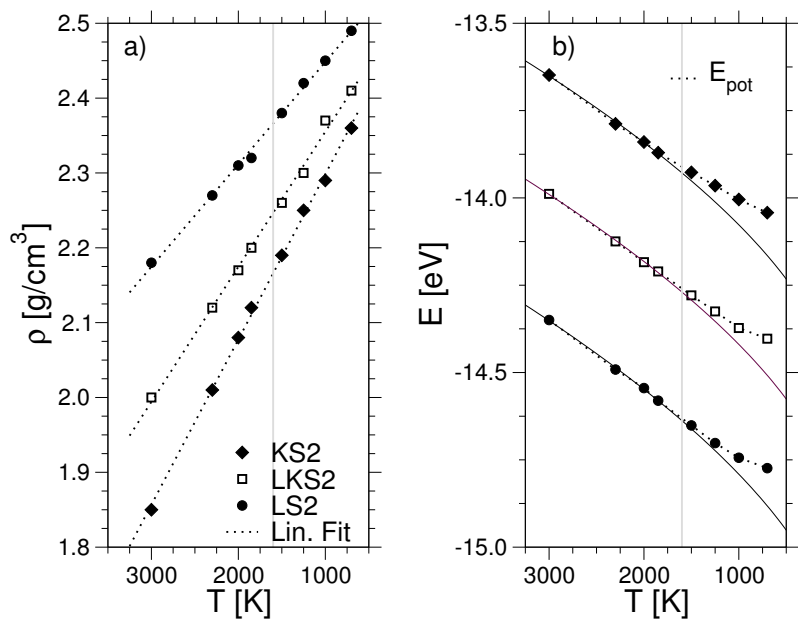
Es sollen zunächst Gemeinsamkeiten und Unterschiede zu den binären Systemen untersucht werden, um dadurch ein besseres mikroskopisches Verständnis der Ionenleitung im allgemeinen und des MAE im speziellen zu erlangen.

## 5.1. Strukturelle Eigenschaften

Auch hier soll die Betrachtung der strukturellen Eigenschaften, wie in Kapitel 3.1, mit der Abhängigkeit von Dichte und potentieller wie totaler Energie von der Temperatur beginnen.

Wie man in Abbildung 5.1 gut erkennt, fügen sich sowohl Dichte als auch die Energien von LKS2 relativ mittig zwischen die Daten der binären Systeme ein. Zudem weisen sie im untersuchten Temperaturbereich ebenfalls eine lineare Abhängigkeit von der Temperatur auf.

**Abbildung 5.1.:** a) Dichte über Temperatur. Die gestrichelten Linien sind ein linearer Fit der Daten. b) Potentielle Energie in Abhängigkeit von der Temperatur. Die durchgezogenen Linien stellen die Vorhersage des Verlaufs der potentiellen Energie nach Rosenfeld and Tarazona [120] dar. Die grauen senkrechten Linien sind eine Abschätzung von  $T_g^{\text{sim}}$  (vgl. Kapitel 3).



Auch bei der potentiellen Energie von LKS2, die zwischen 3000 K und 2000 K gefittete wurde, weicht ab ca. 1600 K von der Vorhersage von Rosenfeld and Tarazona [120] ab (vgl. auch Kapitel 3.1). Es ist also davon auszugehen, daß auch hier das System bei  $T_g^{\text{sim}} \approx 1600$  K aus dem Gleichgewicht fällt.

Die lokale Struktur wird hier zunächst mittels der partiellen Paarkorrelation  $g_{\alpha\beta}(r)$  betrachtet.<sup>1</sup> Eine Übersicht über alle in LKS2 vorkommenden Paarkorrelationen bietet Abbildung 5.2. Einige Unterschiede zu den  $g_{\alpha\beta}(r)$  in den vorher untersuchten binären Systemen sollen im folgenden hervorgehoben werden.

In Abbildung 5.3.a) sind alle Alkali–Sauerstoff Paarkorrelationen der binären und des ternären Systems für eine hohe (2000 K) und eine niedrige (1000 K) Temperatur zusammengestellt. Wie schon zuvor erwähnt, erkennt man deutlich die verschiedenen Längenskalen der NN<sup>2</sup>–Abstände der binären Systeme. Diese setzten sich in LKS2 weiter fort, da sich die Peak– und Minima–Positionen nur marginal bis gar nicht ändern.

Eine Änderung ist jedoch in Höhe und Breite, speziell des 1. NN–Peaks, zu erkennen. Von LS2 nach LKS2 nimmt der erste Peak von  $g_{\text{LiO}}(r)$  an Höhe zu und verbreitert sich. Das bedeutet, daß die Lithium–Ionen anteilig stärker in kleinen Längenskalen lokalisiert sind. Ab dem 4. NN–Peak (hier nicht abgebildet) erkennt man eine schwächere Ausprägung der Nachbarschaft. Bei Kalium–Ionen ist es genau umgekehrt. Dort beobachtet man von KS2 nach LKS2 eine Abnahme des 1. NN–Peaks in Höhe und Breite, dafür aber eine stärkere Korrelation ab dem 3. NN–Peak. Diese Daten würden auf dem ersten Blick die Vermutung nahelegen, daß die Koordination in der ersten Nachbarschaftsschale für Lithium zunimmt und für Kalium abnimmt. Doch hier darf nicht vergessen werden, daß der relative Anteil der einzelnen Ionensorten hier nur jeweils 16.6 mol% beträgt. Darum ist es sinnvoll, erst die Koordinationszahlverteilungen<sup>3</sup> in Abbildung 5.3.b) zu betrachten.

Dort erkennt man, daß sich die Koordinationszahlverteilungen von den binären Systemen hin zu LKS2 im wesentlichen durch den Mittelwert der Verteilung ändert. Bei 1000 K beträgt die mittlere Koordination von Sauerstoff um Lithium in LS2 5.02, Sauerstoff um Lithium in LKS2 4.64, um K in KS2 ist es 7.97 und um K in LKS2 9.69. Das heißt, in LKS2 nimmt die Nachbarschaft von Sauerstoff um Lithium ab, wohingegen die Nachbarschaft von Sauerstoff um Kalium steigt. Der große Unterschied der Sauerstoff–Koordination, den man schon in binären Systemen hatte (5.02 zu 7.97), vergrößert sich hier also noch (4.64 zu 9.69). Die chemische Umgebung um verschiedenartige Alkaliionen ist damit noch inkompatibler, als sie es z.B. in Kapitel 4 war. Daraus ist zu erwarten, daß Sprünge eines Ions auf eine Position, die vorher von einem andersnamigen Ion okkupiert war, sehr unwahrscheinlich sein sollten, da die Umgebung deutlich verschieden von der bevorzugten ist. Dies wird später noch ausführlich besprochen.

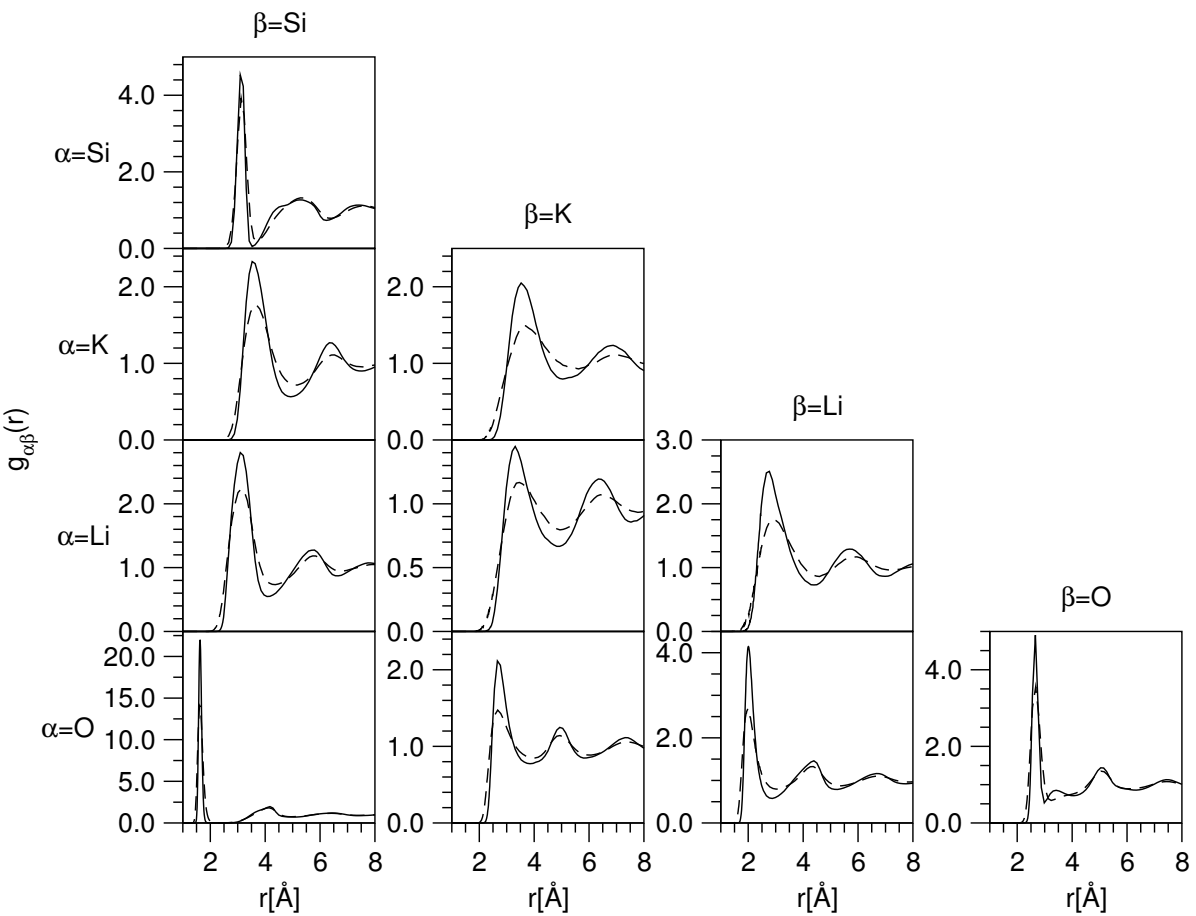
In Kapitel 4 wurde gezeigt, daß Kalium, wenn möglich, mehr Sauerstoff um sich versammelt. Dort hat man gesehen, daß in LS2+1K die mittlere K–O–Koordination bis über 12 ansteigt. Das bedeutet, wenn nicht nur gleichberechtigtes Kalium in System ist, sondern ein anderes Alkaliion noch vorhanden ist, welches Sauerstoff nicht so stark um sich koordinieren kann, dann steigt die K–O–Koordination an. Genau dieser Effekt ist z.B. in Abbildung 4.5 zu sehen.

---

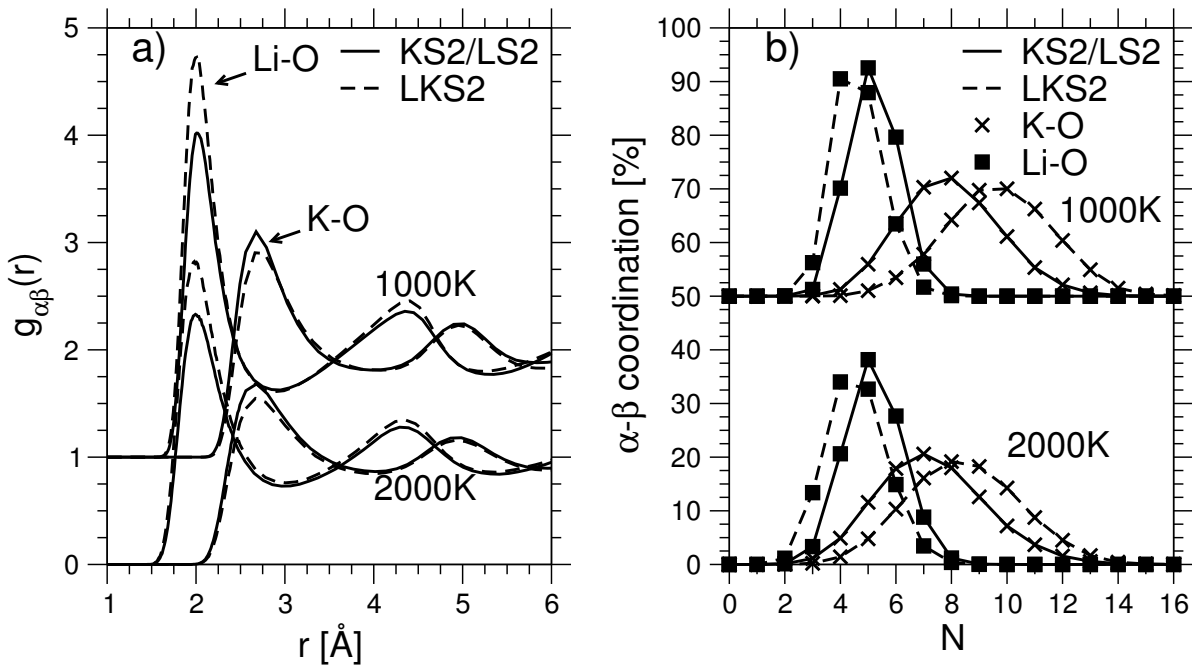
<sup>1</sup>vgl. dazu Kapitel 2.2.1.1.

<sup>2</sup>NN = nächste Nachbarn.

<sup>3</sup>vgl. Kapitel 2.2.1.2.



**Abbildung 5.2:**  $g_{\alpha\beta}(r)$  für alle partiellen Korrelationen in LKS2. Dabei repräsentieren die durchgezogenen Linien 700 K und die gestrichelten Linien die Paar-Korrelationen bei 2300 K.

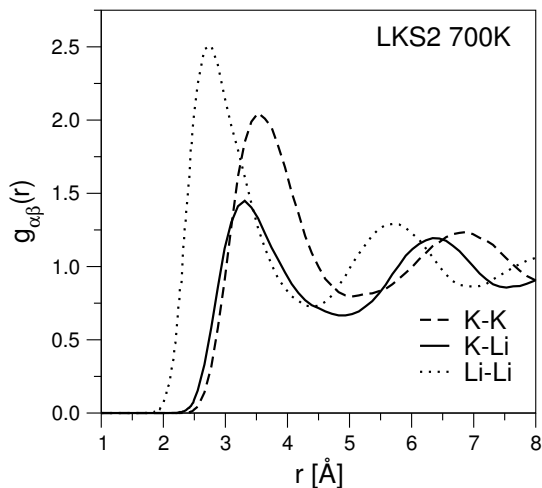


**Abbildung 5.3.:** a) Partielle Paarkorrelationsfunktionen  $g_{\alpha\beta}(r)$  für Li – O und K – O in den binären Systemen KS2 und LS2 sowie für LKS2. Daten für 1000 K um 1 nach oben verschoben. Speziell im ersten NN-Peak erkennt man in LKS2 eine Zunahme der Höhe wie auch eine Verbreiterung für  $g_{\text{LiO}}(r)$ , wohingegen der 1. NN-Peak für  $g_{\text{KO}}(r)$  sogar in Höhe und Breite abnimmt. b) Histogramme der Koordinationszahlen. Daten für 1000 K um 50% nach oben verschoben. Gut zu erkennen ist in LKS2 die Abnahme der Koordination mit Sauerstoff für Lithium, die Kalium–Sauerstoff–Koordination hingegen nimmt zu.

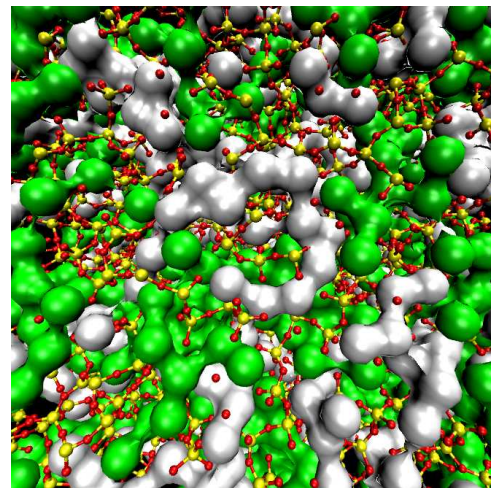
Eine weitere interessante Paarkorrelation in LKS2 ist  $g_{\text{KLi}}(r)$  (Abbildung 5.4). Die NN–Distanz von K–Li liegt dabei mit ca. 3.31 Å zwischen den Maxima von Li–Li (2.77 Å) und K–K (3.53 Å). Das der Verlauf der K–Li–Korrelation fast einer Art Schnittmengenbildung der gleichnamigen Alkali-Korrelationen gleicht, zeigt, daß in dem System häufig Nachbarschaften der Art K–O–Li vorliegen müssen. Das wiederum zeigt, daß wir hier maximal eine Mikrophasenseparation<sup>4</sup> haben können, jedoch keine Phasenseparation. Weitere Details dazu werden später bei der Betrachtung von partiellen Struktur Faktoren erörtert. Doch die Struktur, die sich ausbildet, läßt sich schön anhand eines Schnappschusses der Konfiguration eines LKS2–Systems erklären. In Abbildung 5.5 liegt ein solcher vor. Gut erkennbar ist, daß sowohl Lithium (weiß) als auch Kalium (grün) jeweils ihr eigenes Subnetzwerk ausbilden. Doch sind diese Netzwerke nicht auf größeren Längenskalen getrennt, sondern es finden sich oft Verläufe derart, daß K und Li, über die Vermittlung eines Sauerstoffs, Nachbarn sind.

Bei der Betrachtung der  $g_{\alpha\beta}(r)$  in LKS2 wird also klar, daß die Struktur in Bezug auf Längenskalen sich für kleine Abstände  $r$  nicht stark von der in LS2 oder KS2 unterscheidet. Das impliziert aber, daß die Alkaliionen Ihre Längenskalen aus den binären Systemen auch dem Mischsystem LKS2 „aufprägen“. Dahingehend stellt sich die Frage, ob das auch für intermediäre Längenskalen zutrifft.

<sup>4</sup>Mikrophasenseparation bezeichnet hier einen lokalen Überschuß einer Ionensorte.



**Abbildung 5.4.:** Partielle Paarkorrelationen der Alkaliionen in LKS2 bei 700 K. Gut erkennbar ist, daß die Maxima und Minima von  $g_{\text{KLi}}(r)$  im wesentlichen eine Mittelung dieser Positionen zwischen  $g_{\text{KK}}(r)$  und  $g_{\text{LiLi}}(r)$  darstellt. Dieses Verhalten ist klar, da eine Nachbarschaft über ein vermittelndes O-Atom, also z.B. K–O–Li, von der Länge zwischen K–O–K und Li–O–Li liegen muß.



**Abbildung 5.5.:** Visualisierung einer LKS2-Konfiguration wie in Abbildung 3.12. Gut erkennbar sind die zwei unabhängigen Subnetzwerke von Kalium- (grün) und Lithiumionen (weiß).

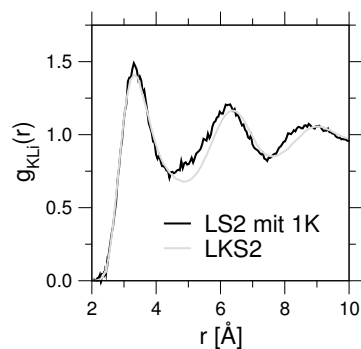
Dazu eignet sich die Auswahl partieller Strukturfaktoren<sup>5</sup>  $S_{\alpha\beta}(q)$  in Abbildung 5.7. Dort sind links die gleichnamigen Alkali–Alkali Korrelationen und rechts die Alkali–Sauerstoff Korrelationen für binäre und das ternäre System aufgetragen. Zudem sind die Korrelationen über alle Alkaliionen aufgetragen. Diese werden dann mit A–A bzw. A–O bezeichnet, wobei  $A = \{\text{K}, \text{Li}\}$ .

Vergleicht man zunächst die K–K, Li–Li sowie die K–O und Li–O Korrelationen zwischen binären und ternären Systemen, so fallen zwei Dinge auf. Erstens sind die Prepeaks<sup>6</sup> in den LKS2–Korrelationen nur noch ansatzweise zu erkennen. Zum zweiten zeigen diese Korrelat-

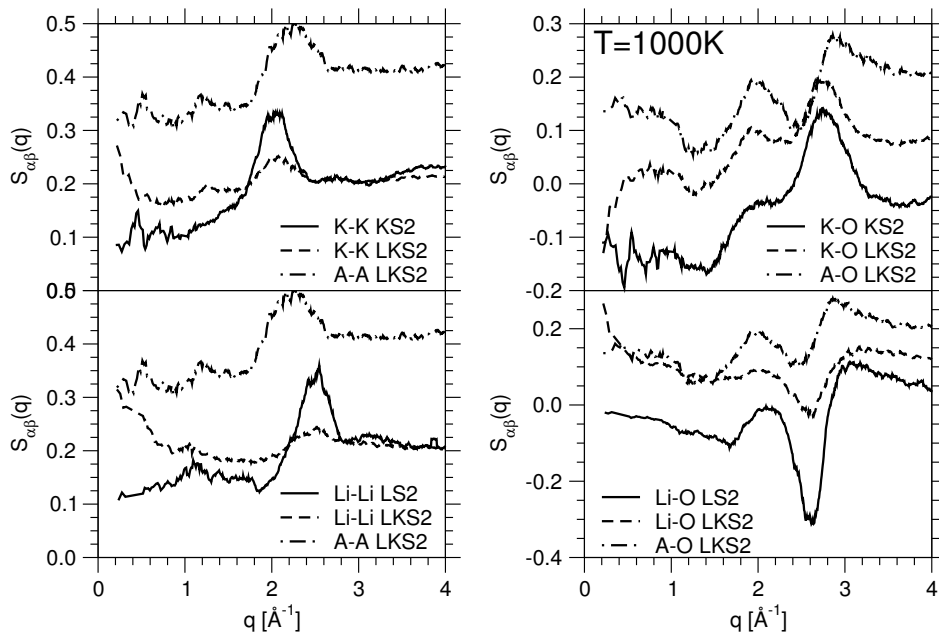
<sup>5</sup>vgl. Kapitel 2.2.1.5.

<sup>6</sup>vgl. Kapitel 3.1 zur Definition von Prepeaks.

**Abbildung 5.6.:** Vergleich von  $g_{\text{KLi}}(r)$  aus LS2 mit einem Fremdkaliumion (LS2+1K) und LKS2. Gut erkennbar ist die gute Übereinstimmung des ersten NN–Peaks. Schon ab dem ersten Minimum unterschieden sich die beiden Paarkorrelationen. Der Grund dafür ist, daß im Fall von LS2+1K die Längenskalen der übernächsten und folgenden Nachbarn nur durch Lithium bestimmt werden. Im Fall von LKS2 ist aber Kalium gleichberechtigt zu Lithium im System vorhanden und prägt dem System seine größere Längenskala auf.







**Abbildung 5.7.:** Vergleich ausgewählter  $S_{\alpha\beta}(q)$  zwischen binären und ternären Systemen. Dabei wurden die gestrichelten Kurven um 0.1 und die gestrichpunkteten Linien um 0.2 nach oben versetzt, um eine bessere Übersichtlichkeit zu erreichen. A bezeichnet hier die Menge aller Alkaliionen, d.h.  $A = \{K, Li\}$ .

ionen für  $q \rightarrow 0$  ein stark kompressibles Verhalten, das die eben geführte Argumentation hinsichtlich der Mikrophasenseparation untermauert. Betrachtet man nun die Korrelationen, in denen nicht zwischen den verschiedenen Alkaliionen unterschieden wird, so findet man die  $q$ -Vektoren, die vorher in beiden binären Systemen beobachtet wurden, wieder. Konkret erkennt man in  $S_{AA}(q)$  Prepeaks bei  $q_{1'} = 1.2 \text{ \AA}^{-1}$  und  $q_{2'} = 0.55 \text{ \AA}^{-1}$ , wobei  $q_{2'}$  eine breite Schulter bis nach  $0.7 \text{ \AA}^{-1}$  aufweist. Damit entspricht der Prepeak  $q_{1'}$  genau dem charakteristischen  $q$ -Vektor  $q_1$  von  $S_{LiLi}(q)$  in LS2 und  $q_{2'}$  vereinigt die  $q$ -Vektoren  $q_{2a/b}$  von  $S_{KK}(q)$  in KS2 in sich. Damit enthält LKS2 sowohl die ausgeprägten Strukturen von LS2 als auch KS2, d.h. man hat es hier mit einer noch einmal komplexeren Struktur zu tun, als im Fall binärer Alkalidisilikate.

Anschaulich wird das Auftreten der Prepeaks auch wieder am Schnappschuß in Abbildung 5.5 klar: Beide Ionensorten haben ihr eigenes Subnetzwerk innerhalb der Matrix (Mikrophasenseparation), diese Subnetzwerke finden sich aber in charakteristischen Abständen zueinander ein, die in den Prepeaks sichtbar sind.

Die Bond-Bond Winkelverteilungen von ternären LKS2 (Abbildung 5.8) verhalten sich qualitativ so, wie in Kapitel 3.1 für binäres KS2 und LS2 beschrieben. Die Unterschiede kann man am besten durch einen direkten Vergleich, hier in Abbildung 5.9, herausarbeiten.

In Abbildung 5.9 sind die Winkelverteilungen der drei Systeme bei einer Temperatur von 700 K einander gegenübergestellt. Den Vergleich kann man zunächst in zwei Blöcke aufteilen: die O- $\beta$ -O Winkelverteilungen und die restlichen Verteilungen.

Bei diesen ist kein dramatischer Unterschied zwischen den Systemen zu erkennen. Einige

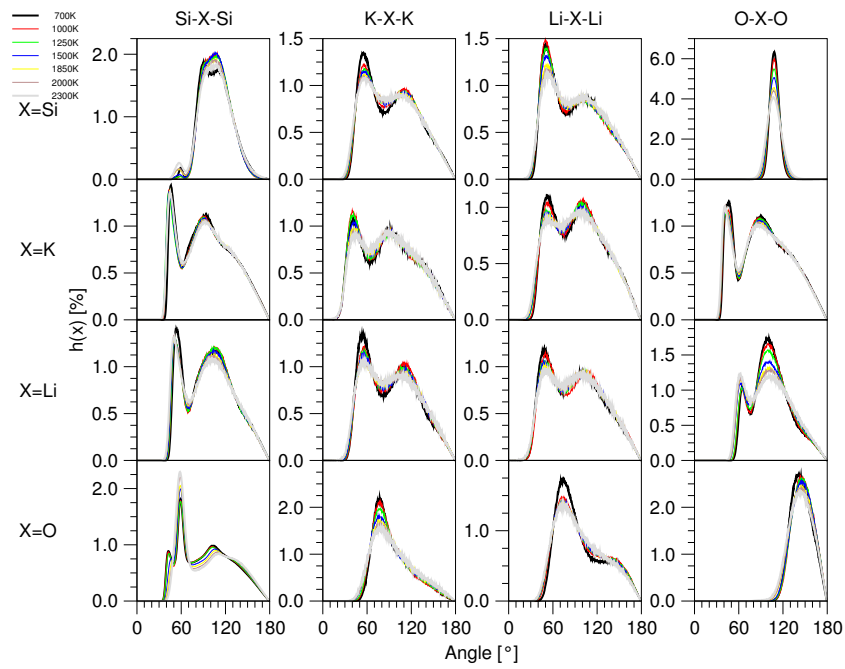


Abbildung 5.8.: LKS2 Bond-Bond Winkelverteilung.

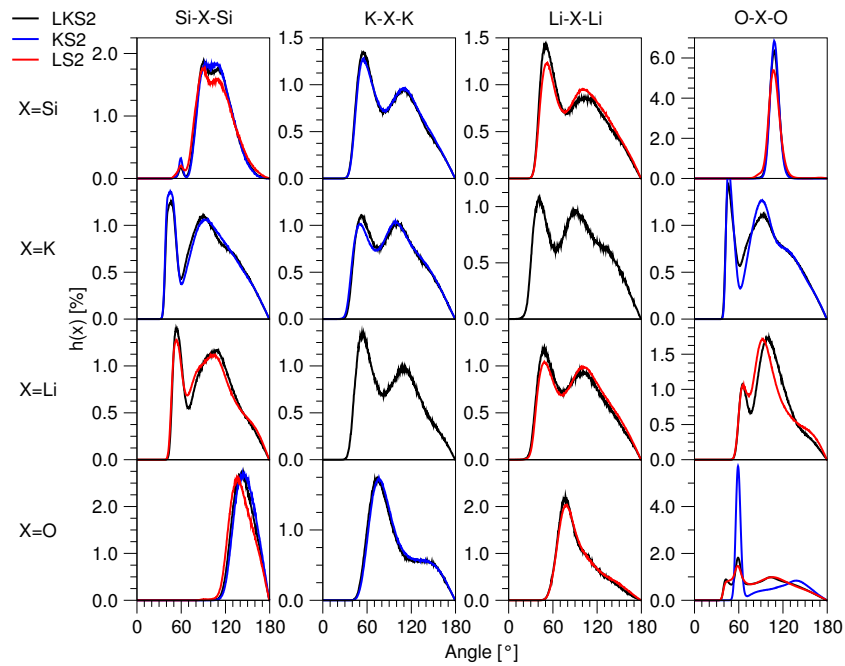


Abbildung 5.9.: Bond-Bond Winkelverteilungen von LKS2 (schwarz), KS2 (blau) und LS2 (rot) bei 700 K.

Peaks und Minima sind etwas ausgeprägter (z.B. bei Si–Si–Si oder Li–Si–Li), die Positionen der Winkel etwas verschoben (z.B. bei Si–O–Si), aber im wesentlichen stimmen die Verteilungen wenigstens qualitativ miteinander überein.

Dasselbe trifft auch im wesentlichen für die O– $\beta$ –O-Verteilungen zu, wenn auch etwas stärker ausgeprägt als bei allen anderen Winkelverteilungen. Erwähnenswert ist hier aber speziell die O–O–O Winkelverteilung. Deren Verlauf entspricht nämlich, wie z.B. bei Si–Si–Si oder Si–O–Si, eher dem, den man in KS2 antreffen würde. O–O–O ist die einzige Winkelverteilung in LKS2, die im Verlauf im wesentlichen dem Verhalten in LS2 entspricht.

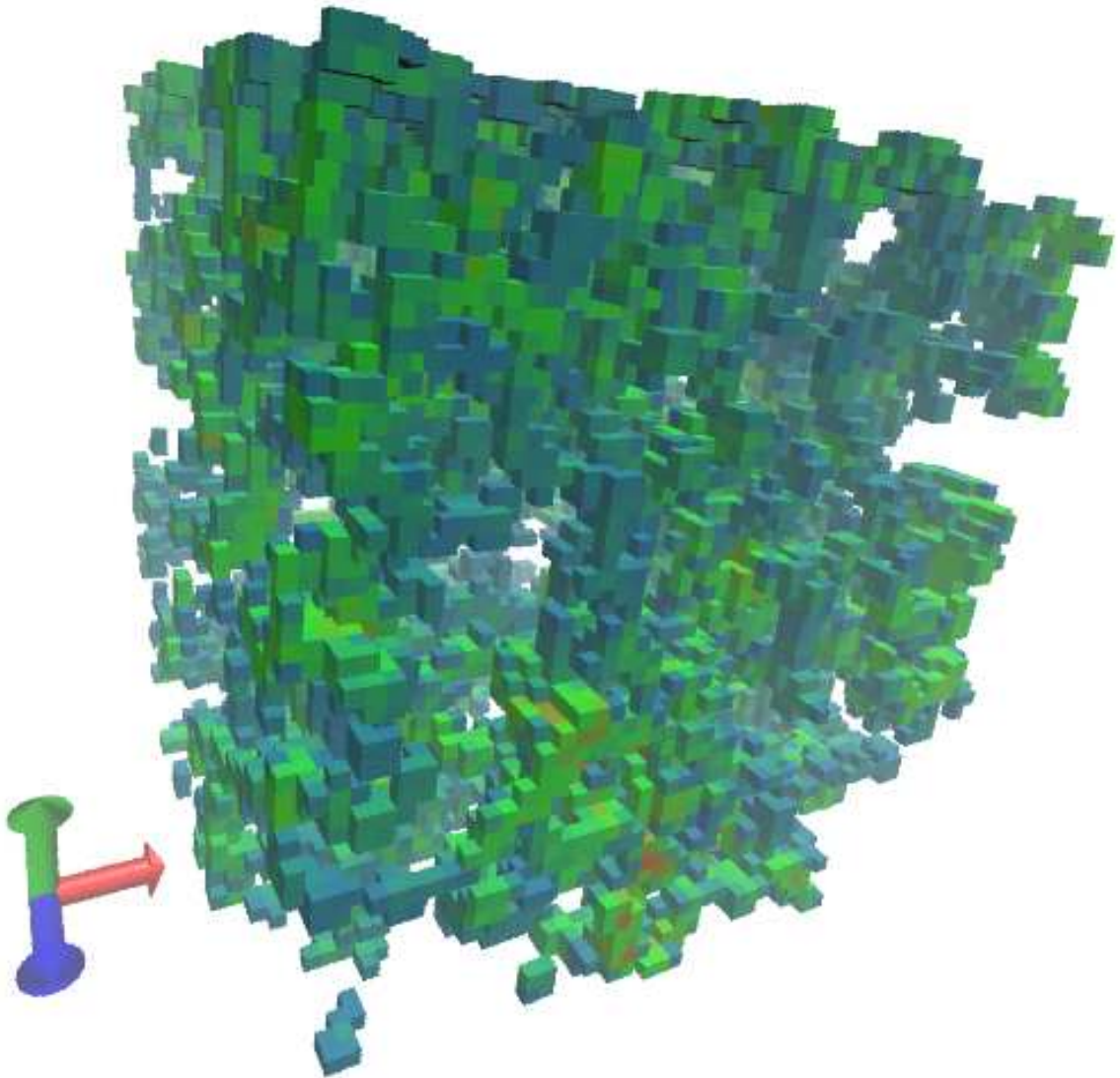
Zur Betrachtung der Statik der Subnetzwerke wurde noch ein weiteres Hilfsmittel herangezogen. Dafür wurde die Simulationsbox in kleine kubische Subboxen mit der Kantenlänge  $l$  unterteilt (in Abbildung 5.10 ist  $l = 0.5 \text{ \AA}$ ). Dann wurde über einen Zeitraum  $t$  (in dieser Untersuchung jeweils  $t = 78 \text{ ps}$ ) mit einer Zeitauflösung  $\Delta t$  (hier  $\Delta t = 0.16 \text{ ps}$ ) geprüft, ob sich Alkaliionen in einer Subbox befinden. Wurde die Subbox noch nicht von einem Alkaliion „besucht“, so wird diese Subbox markiert. Wurde die Subbox hingegen schon zuvor von einem Alkaliion besucht, so wird geprüft, ob es sich um dasselbe Ion handelt, oder um ein neues. Handelt es sich um ein neues, so wird die Zahl der verschiedenen Ionen, die diese Subbox besucht haben, inkrementiert. Visuell wird dies in Abbildung 5.10 durch die Farbskala dargestellt, in der Dunkelblau für wenige Ionenbesuche, dunkelrot für häufige Ionenbesuche steht. Von der Häufigkeit der verschiedenen Ionen, die eine Box besuchen, wird im folgenden nicht mehr die Rede sein, da sich aus dieser Eigenschaft keine weiteren Erkenntnisse ergaben. Daher basieren alle folgende Analysen mittels markierter Subboxen darauf, daß mindestens ein Alkaliion sie besucht hat.

In Abbildung 5.11 sieht man die Entwicklung des Volumens (1000 K), daß von den Subboxen nach einer Zeit  $t$  eingenommen wird ( $V_\alpha(t)$ ). Die Größe ähnelt stark den MSD's. Für die Si-Atome erkennt man für große Zeiten nur noch einen kleinen Zuwachs des besuchten Volumens. Da es für lange Zeiten immer noch einen Zuwachs gibt, obwohl die Si-Atome nicht diffusiv werden, ist wohl einerseits auf die Alterungseffekte zurückzuführen, die in Kapitel 3.3 besprochen werden, zum anderen und wohl hauptsächlich aber auf den Unterschied in der Definition der Größen. Im MSD wird nur der Abstandsvektor betrachtet, bei  $V_\alpha(t)$  wird jedoch jede Subbox markiert, in die das Teilchen eintritt. Wenn also ein Teilchen aus dem ballistischen Regime, in dem es sich geradlinig fortbewegt hat und somit  $\propto t$  Subboxen markiert hat, austritt, so setzt bei niedrigen Temperaturen der Käfigeffekt<sup>7</sup> ein. Das bedeutet aber nicht, daß, bildlich gesprochen, das Teilchen an den Nachbarn, die es zuerst erreicht, kleben bleibt, sondern, daß das Teilchen das Volumen im Käfig „erkundet“ und somit selbst in der  $\beta$ -Relaxation noch lange zum Zuwachs von  $V_\alpha(t)$  beiträgt. Generell ist das Verhalten von  $V_{\text{Si}}(t)$  in LKS2 zwischen dem Verhalten von  $V_{\text{Si}}(t)$  in KS2 und LS2 angesiedelt, wobei es etwas mehr dem Verhalten in LS2 entspricht.

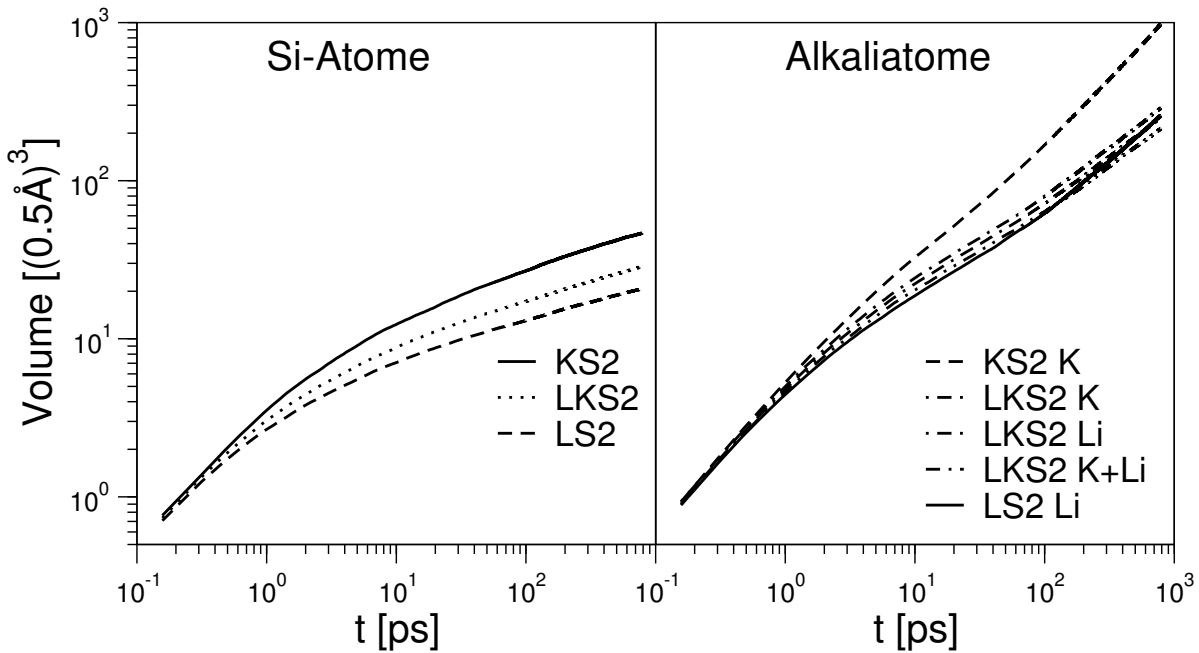
Rechts in Abbildung 5.11 sind die  $V_\alpha(t)$  für die Alkaliionen zu sehen. Das Verhalten entspricht dem eben beschriebenen, allerdings verlassen die Alkaliionen den Bereich der  $\beta$ -Relaxation und gehen in das diffusive Regime über, was im Verhalten von  $V_\alpha(t)$  auch zu beobachten ist. Interessant ist dabei, das selbst Kalium in LKS2 vom Verhalten sich eher an Li in LS2 orientiert, als an K in KS2. In diesem Fall zeigt das Lithium einen starken Einfluß auf die Dynamik der Kaliumionen.

---

<sup>7</sup>vgl. Kapitel 1.4.



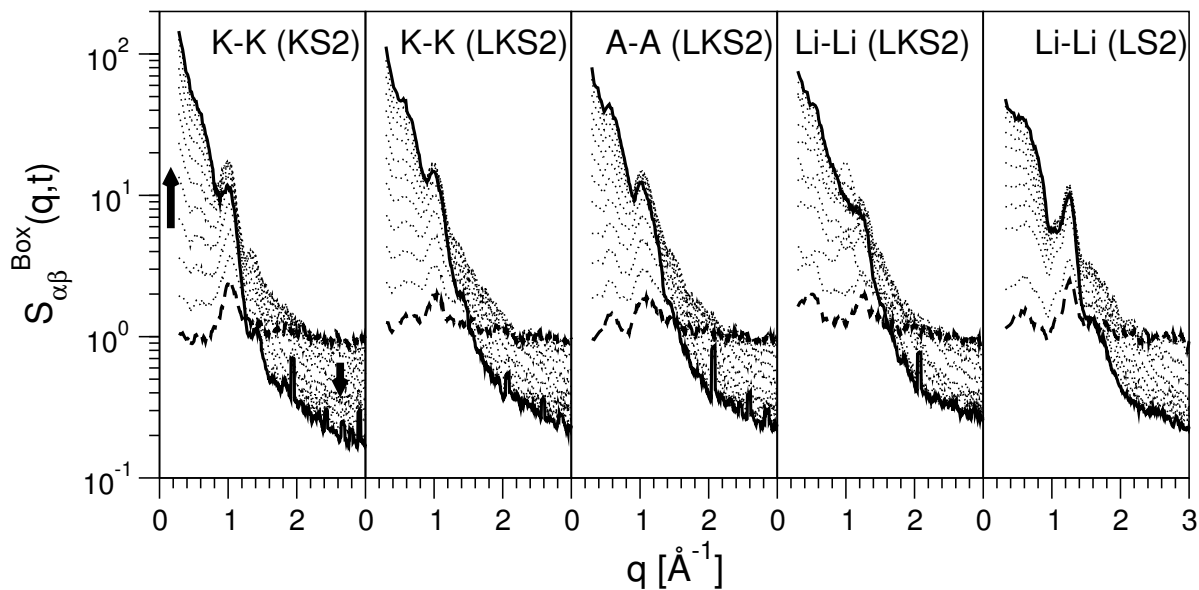
**Abbildung 5.10.:** Visualisierung der Aufteilung einer Simulationsbox in kubische Subboxen mit Kantenlänge  $0.5 \text{ \AA}$ . Hier als Beispiel ein KS2-System bei 1000 K. Eingefärbt sind diejenigen Subboxen, die mindestens 10 mal von verschiedenen Kaliumionen besucht wurden. Dabei gibt die Färbung an, wieviele verschiedene Kaliumionen die Subbox besucht haben. Eine geringe Anzahl wird repräsentiert durch dunkelblau (hier 10), rot eine hohe Anzahl (48) von verschiedenen Ionen, die die Subbox im Simulationszeitraum besuchten.



**Abbildung 5.11.:** Mittleres dynamisches Volumen eines Si- oder Alaklilteilchens in  $(0.5 \text{ \AA})^3$ -Blöcken (bei 1000 K).

Durch die zu verschiedenen Zeiten markierten Subboxen ist man jetzt in der Lage, nicht nur einzelne Teilchenpositionen wie in  $S_{\alpha\beta}(q)$  auszuwerten, sondern ganze Raumbereiche in die strukturelle Untersuchung einzubeziehen. Dazu sollen jetzt die partiellen Strukturfaktoren  $S_{\alpha\beta}^{\text{Box}}(q, t)$  betrachtet werden. Jede von einem Ion bis zum Zeitpunkt  $t$  besuchte Subbox repräsentiert dabei einen Punkt im Raum. Mit dieser Punkteverteilung wird dann der partielle statische Strukturfaktor wie in Kapitel 2.2.1.5 beschrieben berechnet.

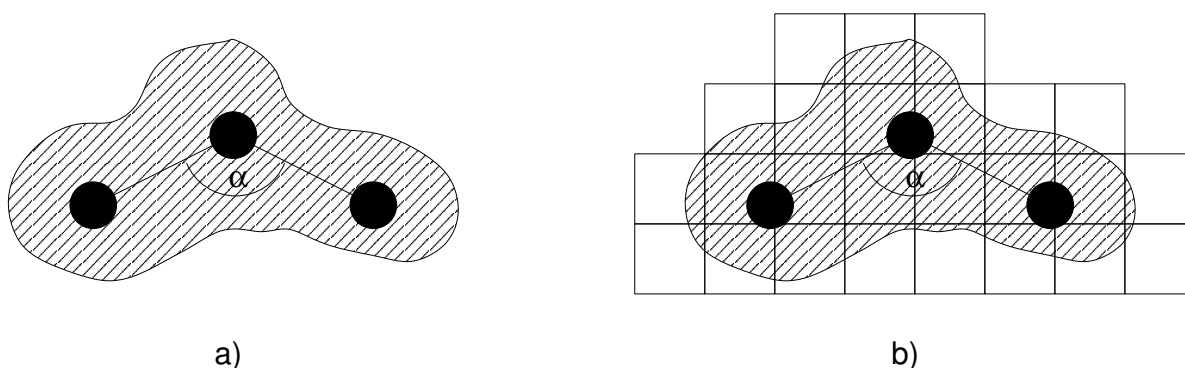
Einen Überblick über die  $S_{\alpha\beta}^{\text{Box}}(q, t)$  der Alkaliionen bei 1000 K liefert Abbildung 5.12. Für lange Simulationszeiten erkennt man in allen  $S_{\alpha\beta}^{\text{Box}}(q, t)$  für  $q \lesssim 1.5 \text{ \AA}^{-1}$  eine Zunahme der Amplitude für  $q \rightarrow 0$ , wohingegen für größere  $q$  die Amplitude deutlich abnimmt. Das zeigt wieder, wie vorher besprochen, daß in den Systemen eine Mikrophasenseparation stattfindet. Die FSDP, die in  $S_{\alpha\beta}(q)$  auftreten, beobachtet man hier in den  $S_{\alpha\beta}^{\text{Box}}(q, t)$  nicht, denn diese beschreiben die NN-Abstände. Dadurch, daß man nicht einzelne Teilchenkonfigurationen, sondern „kumulierte“ Konfiguration betrachtet, treten die NN-Abstände gar nicht mehr auf. Betrachtet man nun die Peaks (für längere Zeiten gehen diese wegen der Steigung für  $q \rightarrow 0$  teils in Schultern über) der  $S_{\alpha\beta}^{\text{Box}}(q, t)$ , so erkennt man zunächst die bekannten Prepeaks aus  $S_{\alpha\beta}(q)$  wieder: Für K-K in KS2 findet sich im Bereich von  $0.5 - 0.7 \text{ \AA}^{-1}$  ein kleiner Prepeak (Schulter) wieder. Diese findet man auch in K-K und A-A in LKS2, analog zu den  $q$ -Werten  $q_{2a,b}$  für  $S_{\alpha\beta}(q)$  zuvor. Ähnlich das Bild für Lithium. Der Prepeak  $q_1$  bei ca.  $1.2 \text{ \AA}^{-1}$  findet sich auch hier für die  $S_{\alpha\beta}^{\text{Box}}(q, t)$  für Li-Li in LS2 und LKS2, sowie in A-A in LKS2 wieder. Soweit die Übereinstimmung von  $S_{\alpha\beta}^{\text{Box}}(q, t)$  und  $S_{\alpha\beta}(q)$ . Es tritt aber auch in  $S_{\alpha\beta}^{\text{Box}}(q, t)$  Ordnung auf Längenskalen auf, die man in  $S_{\alpha\beta}(q)$  nicht gesehen hat. Bei allen Größen, in denen Kalium beteiligt ist, treten Peaks (oder verbreiterte Peaks (Schultern)) bei ca.  $1 \text{ \AA}^{-1}$  auf und in allen



**Abbildung 5.12.:**  $S_{\alpha\beta}^{\text{Box}}(q, t)$  von Kanalnetzwerken. Dazu wurde die Simulationsbox, wie im Text beschrieben, aufgeteilt. Während eines 786 ps dauernden Simulationslaufs wurden die Subboxen markiert, in denen sich die beteiligten Alkaliionen  $\{\alpha, \beta\}$  aufgehalten haben. Diese Subboxen wurden dann als Punktteilchen betrachtet, um damit den Strukturfaktor des Kanalnetzwerkes zu den Zeiten 0.31 ps (gestrichelte Linie) bis 643.8 ps (durchgezogene Linie) bestimmen (die gepunkteten Linien repräsentieren die dazwischenliegenden Zeiten, die jeweils durch Verdopplung der vorherigen Zeit gewählt sind, also 0.62 ps, 1.24 ps usw.). Gut zu erkennen ist der stark von 0 abweichende Verlauf für  $q \rightarrow 0$  bei zunehmenden Simulationszeiten. Bei den Kanälen handelt es sich ja auch nicht um eine homogene Verteilung, sondern schon um einen ersten Schritt der Separation vom Silikat bzw. der anderen Alkali-komponente.

Systemen mit Beteiligung von Lithium sieht man Peaks bei ca.  $0.5 \text{ \AA}^{-1}$ .

Der Ursprung dieser Strukturbildung wurde in dieser Arbeit nicht näher untersucht, es gibt jedoch zwei Vermutungen, weshalb sie auftritt. Die eine Möglichkeit ist, daß diese neue Längenskalen die charakteristische Größe kleiner, knollenartiger Ansammlungen von Alkaliionen

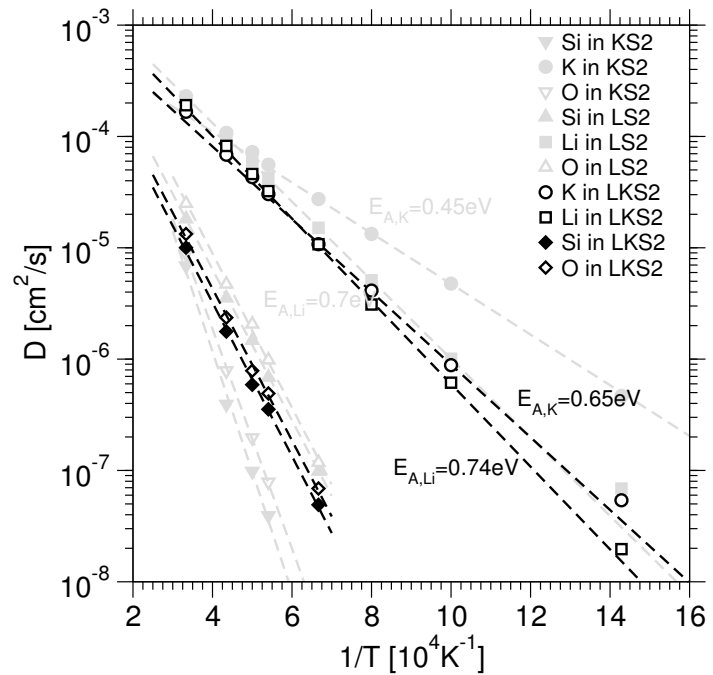


**Abbildung 5.13.:** Illustration zum Text Seite 103.

in der Matrix sind. Dieses Knollen treten immer wieder mal zwischen den Kanälen auf. Die andere Möglichkeit ist, daß es sich um eine charakteristische Länge eines Kanalstücks handelt. Dies soll am Beispiel von einer Kette aus drei Kaliumionen in KS2 und LKS2 erklärt werden. In Abbildung 5.13a) sieht man diese Kette. Die Winkelverteilung liefert zwei Peaks für die K–K–K–Bondwinkelverteilung, einer davon liegt bei  $\alpha \approx 100^\circ$ , der NN–Abstand von K–K beträgt ca. 3.5 Å. Berechnet man nun aufgrund dieser Wertung die längste Ausdehnung  $l_{K-K-K}$  dieser Dreierkette, so erhält man  $l_{K-K-K} = 5.36 \text{ Å}$ . Das ergibt aber noch kein  $q \approx 1 \text{ Å}^{-1}$ . Um das zu verstehen, muß man Abbildung 5.13b) betrachten. Die schraffierte Fläche ist eine zweidimensionale Projektion des Raums, den die Teilchen, die gerade in den drei Plätzen sitzen, durch Bewegung um ihre Ruheposition durchlaufen. Auf diese schraffierte Fläche wurde jetzt ein (grobes) Subboxenraster aufgelegt, um zu verdeutlichen, wie die Boxen liegen, die durch die Teilchenoszillationen markiert werden. Man kann sich leicht überlegen, daß die Längen, die durch ein diskretes Raster dieser Art gemessen werden, immer die realen Ausdehnungen überschätzt. Der mittlere Fehler in jede Raumrichtung eines Subboxenrasters auf Basis kartesischer Koordinaten ist also  $l/2$ . Mit  $l = 0.5 \text{ Å}$  kann man also abschätzen, daß die durch die Subboxenmethode gemessene Länge  $l_{K-K-K}^{\text{Box}} \gtrsim l_{K-K-K} + 2 \cdot 0.25 \text{ Å} = 5.86 \text{ Å}$  ist. Das entspricht einem  $q$ -Vektor von  $1.07 \text{ Å}^{-1}$ , also der Längenskala, die in den  $S_{KK}^{\text{Box}}(q, t)$  auftritt. Die Vermutung soll hier lauten, daß im (Sub)Netzwerk der Kaliumionen solche Dreierkettenelemente häufig vorkommen. Das heißt nicht, daß sie alleine in der Matrix hängen, sondern daß das nächste Element des Netzwerks in einem solchen Winkel dranhängt, daß so keine noch größeren Längenskalen gemessen werden.

Abschließend kann man zusammenfassen, daß die Struktur von LKS2 alle charakteristischen Längenskalen von LS2 und KS2 aufweist. Es stellt sich also eine komplexe Struktur mit zwei perkolierenden Subnetzwerken für die Alkaliionen ein. Auch die Winkelverteilungen zeigen im wesentlichen dieselben Merkmale, wie die der binären Systeme. Die Unterschiede in der chemischen Umgebung der Teilchen vergrößern sich allerdings beim Übergang vom binären ins ternäre System noch. Zudem kann gezeigt werden, daß durch eine Subboxenanalyse Längenskalen gemessen werden, die in den partiellen Struktur Faktoren zuvor noch nicht zu sehen waren.

**Abbildung 5.14.:** Arrheniusplot der Diffusionskonstanten analog zu Abbildung 3.16. Die Daten aus dieser Abbildung sind hier in grau dargestellt. In schwarz hinzugefügt wurden die Daten für das ternäre System LKS2.



## 5.2. Dynamische Eigenschaften und der MAE

In diesem Teil der Arbeit soll nun der kollektive<sup>8</sup> MAE in ternären Alkalidisilikaten genauer betrachtet werden. Zunächst muß die Frage beantwortet werden, ob das Potential überhaupt den MAE, so wie man ihn aus den Experimenten kennt (vgl. Abbildung 1.2 und Abbildung 4.1), reproduziert. Ist das der Fall, so ist die Frage, welche Unterschiede oder Gemeinsamkeiten mit den in dieser Arbeit bereits in Kapitel 3 untersuchten binären Alkalidisilikaten bestehen, bzw. inwiefern man die in Kapitel 4 gemachten Beobachtungen mit dem kollektiven MAE in Verbindung bringen kann.

### 5.2.1. Auftreten und Ausprägung des MAE

Zur Beantwortung der ersten Frage, ob das verwendete Potentialmodell überhaupt in der Lage ist, den MAE zu reproduzieren, werden zunächst einmal die Diffusionskonstanten aufgenommen. Die Resultate sind in einem Arrheniusplot in Abbildung 5.14 als schwarze Eintragungen

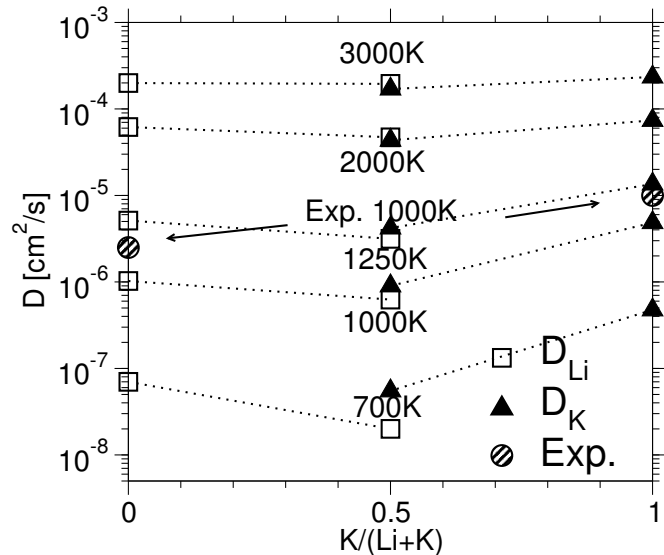
<sup>8</sup>Der Ausdruck „kollektiv“ soll ausdrücken, daß man hier nicht den MAE für eine einzelne Ionensorte betrachtet (wie in Kapitel 4), sondern für alle Komponenten zusammen.

System	$E_{A,K}$ [eV]	$E_{A,Li}$ [eV]
KS2	0.45	-
LKS2	0.65	0.74
LS2	-	0.7

**Tabelle 5.1.:** Übersicht der Aktivierungsenergien aus der Simulation.



**Abbildung 5.15.:** Übersicht über die Diffusionskonstanten der Systeme LS2, LKS2 und KS2, aufgetragen über den Anteil der Ionenspezies. Das bedeutet  $K/(Li+K)=0$  ist das System LS2,  $K/(Li+K)=0.5$  ist LKS2 und  $K/(Li+K)=1$  ist KS2. Bei 3000 K, in der Schmelze, ist kein MAE erkennbar, für fallende Temperaturen bildet sich bei  $x = 0.5$  ein deutliches Minimum aus. Der Unterschied in der Größe der Selbstdiffusion von Kalium in KS2 und LKS2 beträgt bei 700 K eine Größenordnung.

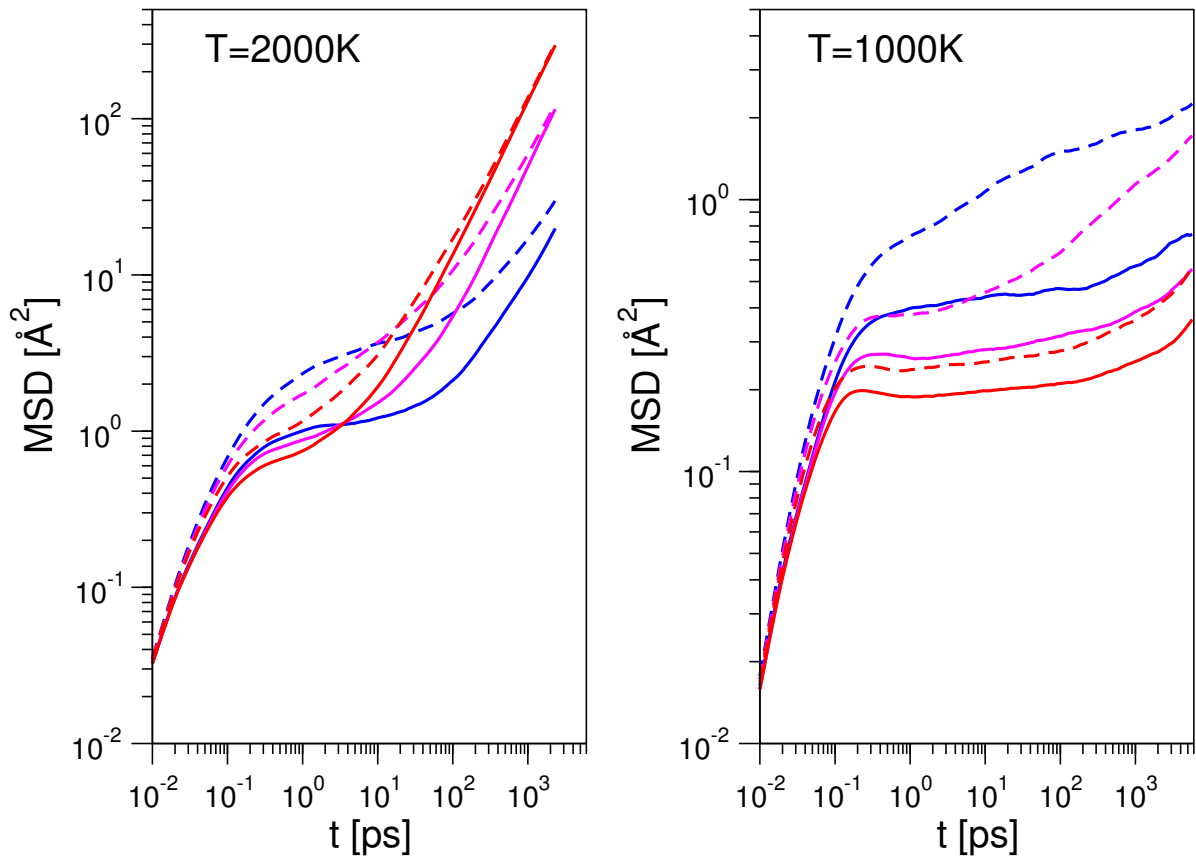


zu sehen. In grau sind zum Vergleich die Diffusionskonstanten der binären Alkalisilikate LS2 und KS2 aufgetragen. Mittels Gleichung (3.1) kann man nun auch die Aktivierungsenergien<sup>9</sup> von Li und K in LKS2 bestimmen und sie mit den Aktivierungsenergien der binären Systeme vergleichen (Tabelle 5.1). Man sieht sofort, daß die Aktivierungsenergien im ternären System für die jeweilige Alkaliionensorte höher sind, als sie es in den binären waren. Besonders groß fällt der Unterschied für Kalium mit  $\Delta E_A = 0.2 \text{ eV}$  aus, was einer Steigerung der Aktivierungsenergie um 45 % entspricht.

Um festzustellen, um welche Größenordnung die Diffusionskonstanten niedriger liegen, wird eine Auftragung der Diffusionskonstanten über die Konzentration der Ionensorten gewählt. Tatsächlich erkennt man in Abbildung 5.15, daß für niedrigere Temperaturen die Diffusionskonstante für LKS2 ( $K/(K + Li) = 0.5$ ) relativ zu den Diffusionskonstanten der binären Systeme ( $K/(K + Li) = 0$  und  $K/(K + Li) = 1$ ) immer weiter absinkt. Bei 3000 K erkennt man noch keinen Effekt, aber bei 700 K beträgt der Unterschied der Diffusionskonstanten zwischen binären und ternären System für Lithium ca. eine halbe, für Kalium aber schon eine ganze Größenordnung.

Eine interessante Beobachtung gibt es noch zur Dynamik der Sauerstoffe auf intermediären Zeitskalen. Die Diffusionskonstanten, wie sie z.B. in Abbildung 5.14 für Sauerstoff aufgetragen sind, sind immer über alle Sauerstoffe im System gemittelt. Gibt es aber vielleicht Unterschiede zwischen den in der Matrix eingebundenen O-Atomen und den sowohl mit Si als auch mit Alkaliionen koordinierten O-Atomen? Um diese Frage zu klären, wurden für 2000 K und 1000 K die MSD's der Sauerstoffatome aufgenommen, jedoch unterschieden nach dem Kriterium, ob sie einfach oder doppelt mit Si koordiniert sind. Die Ergebnisse dieser Auswertung finden sich in Abbildung 5.16. Man erkennt sofort, daß für kurze Zeiten (ballistisches Regime) und für lange Zeiten (diffusives Regime) das Verhalten der Sauerstoffe unabhängig von der Koordination mit Si oder Alkaliionen ist. im Zeitraum dazwischen, in der  $\beta$ -Relaxation allerdings erkennt man einen deutlichen Unterschied in der Lokalisierung der Sauerstoffatome. Die in der Matrix eingebundenen, zweifach mit Si koordinierten O-Atome zeigen einen

<sup>9</sup>Zur Bestimmung der Aktivierungsenergien wurde ein Arrheniusgesetz im Bereich zwischen 1500 K und 1000 K angefitet.



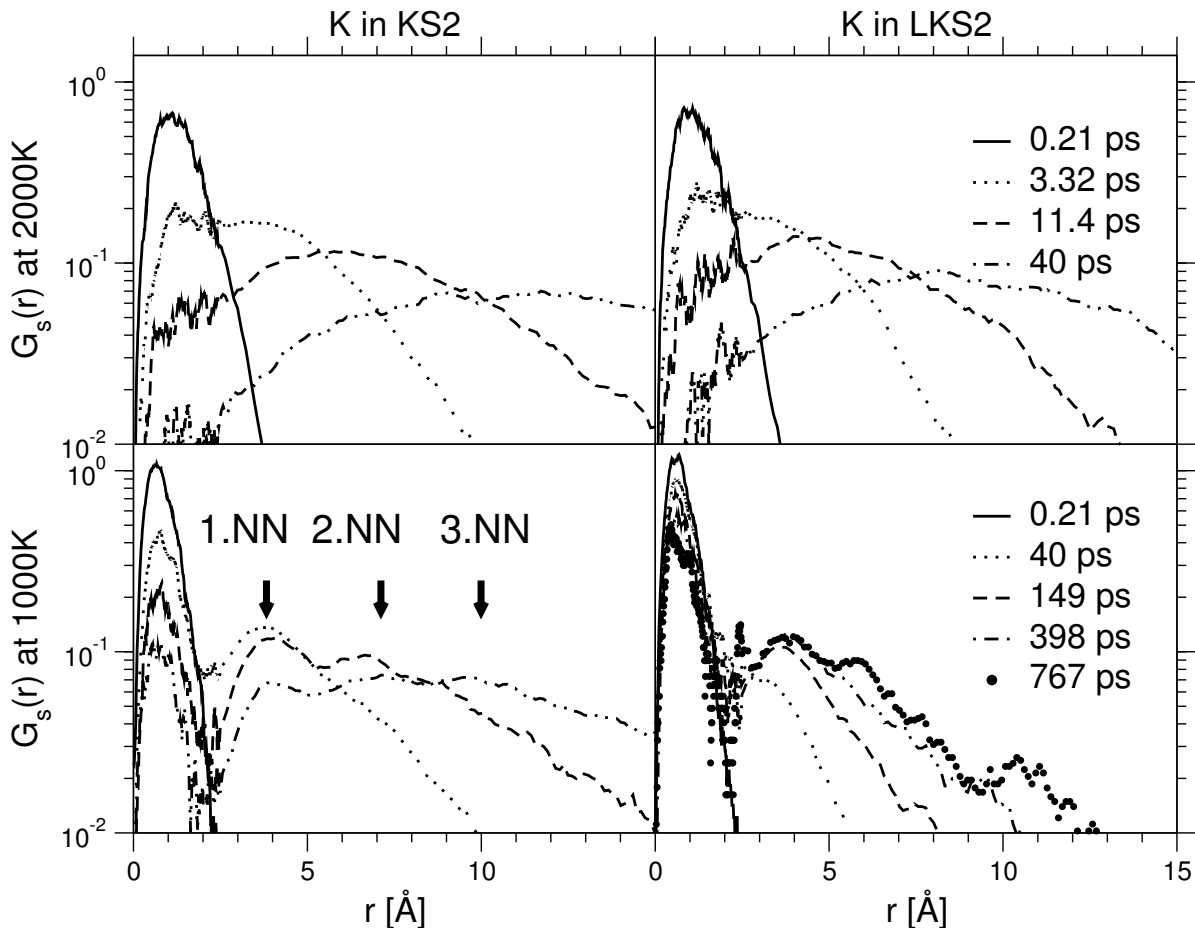
**Abbildung 5.16.:** MSD's von einfach (gestrichelte Linien) und zweifach (durchgezogene Linien) mit Silizium koordiniertem Sauerstoff in den Systemen KS2 (blau), LKS2 (magenta) und LS2 (rot).

deutlich niedrigeren Plateauwert, als es für die O-Atome der Fall ist, die auch mit Alkaliionen koordiniert sind. Das Verhalten ist auch leicht zu verstehen, da zweifach mit Si koordinierter Sauerstoff nicht die Bewegungsfreiheit hat, die ein Sauerstoffatom aufweist, das in den alkalioxidreichen Teil des Systems hineinragt, und durch Winkelveränderung der Bindung mit verschiedenen Alkaliionen koordiniert sein kann.

Noch ist allerdings nicht geklärt, ob man es hier wirklich mit dem MAE zu tun hat. Dazu nun folgende Betrachtung: Aus Kapitel 5.1 weiß man, daß sich genau wie in binären Systemen für jede Alkaliionensorte ein eigenes, perkolierendes Subnetzwerk bevorzugter Ionenplätze ausbildet, in dem die Alkaliionen sich bewegen. Mittels dem Selbstanteil der van-Hove-Korrelationsfunktion (Abbildung 5.17) kann man auch zeigen, daß auch die Bewegung innerhalb dieser Subnetzwerke genauso mittels Hopping zwischen den Alkaliplätzen stattfindet, wie es auch in den binären Systemen der Fall ist.<sup>10</sup> Jetzt kann man vermuten, daß das Potential vielleicht nur das Verhalten reproduziert, wie es im Myuller-Modell<sup>11</sup> angenommen wird, nämlich, daß die Ionen nur und ausschließlich innerhalb ihres eigenen Subnetzwerks

<sup>10</sup>Bis auf die Tatsache, daß man beim Vergleich der  $G_s(r, t)$  von Kalium in Abbildung 5.17 bei 1000 K eine zeitliche Verzögerung von ca. einer Größenordnung erkennt, wohingegen bei 2000 K die  $G_s(r, t)$  für gleiche Zeiten in KS2 und LKS2 fast identisch sind.

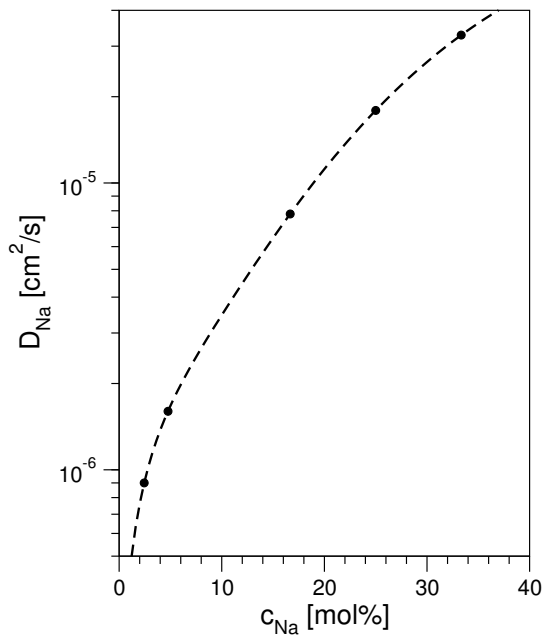
<sup>11</sup>vgl. Kapitel 1.3.



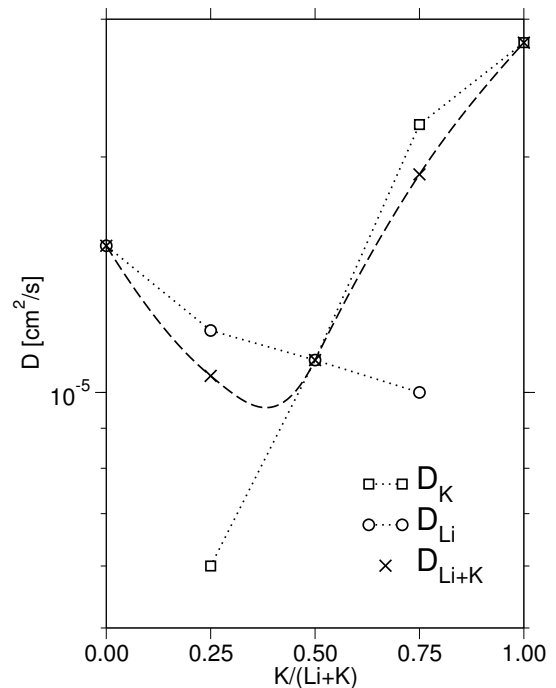
**Abbildung 5.17.:** Selbstanteil der Van-Hove-Korrelationsfunktion für KS2 (wie bereits in Abbildung 3.17 beschrieben) und für Kalium in LKS2. Qualitativ zeigt sich hier das gleiche Verhalten in beiden Systemen. Bei 2000 K bezieht sich das auch auf das zeitliche Verhalten. Bei 1000 K jedoch erkennt man eine zeitliche Verzögerung im Mischsystem. Dort hängt die Dynamik der K-Atome der im binären System ungefähr um einen Faktor 10 nach.

verkehren, ohne den Einfluß der anderen Alkaliionensorte zu spüren. Dann würden sich die Alkaliionen so verhalten, als ob sie in einem binärem Silikat wären, allerdings mit niedrigerer molarer Konzentration. Denn dann sinken die Diffusionskonstanten auch ab. Das wäre aber verwunderlich, da schon in Kapitel 4 gezeigt wurde, daß die verschiedenartigen Alkaliionen einen Einfluß aufeinander haben.

Wäre das aber der Fall, daß man nur eine Verlangsamung der Dynamik aufgrund einer Konzentrationsänderung im eben beschriebenen Sinn beobachtet, dann erwartet man in logarithmischer Auftragung der Diffusionskonstanten über der Konzentration einen konkaven Verlauf, wie er in Abbildung 5.18 für verschiedene Natriumsilikate zu sehen ist. Um Aussagen über das Krümmungsverhalten treffen zu können, wurden zwei weitere Konzentrationsmischungen simuliert,  $0.75(\text{Li}_2\text{O}) 0.25(\text{K}_2\text{O}) 2(\text{SiO}_2)$  und  $0.25(\text{Li}_2\text{O}) 0.75(\text{K}_2\text{O}) 2(\text{SiO}_2)$  bei 1500 K. Diese wurden dann zusammen mit den Daten von LS2, KS2 und LKS2 in Abbildung 5.19 eingetragen. Tatsächlich erkennt man für Lithium ein deutlich konvexes Verhalten, wohinge-



**Abbildung 5.18.:** Na-Diffusionsdaten (Punkte) für verschiedene Konzentrationen bei  $T = 2500$  K und  $\rho = 2.37$  g/cm<sup>3</sup>. Die gestrichelte Kurve ist eine Fitkurve der Datenpunkte. Gut zu erkennen ist der in logarithmischer Auftragung konkave Verlauf der Diffusionskonstanten in Abhängigkeit von der Na-Konzentration  $c_{\text{Na}}$ . (Zur Verfügung gestellt von Horbach [59]).



**Abbildung 5.19.:** Diffusionskonstanten bei 1500 K. Aufgetragen sind Punkte aus LS2, LKS2, KS2 und zusätzlich berechnete Punkte für  $x = 0.25$  und  $x = 0.75$ , wobei  $x$  den Konzentrationsanteil in  $(1 - x)(\text{Li}_2\text{O}) x(\text{K}_2\text{O}) 2(\text{SiO}_2)$  steuert. Die gestrichelte Linie ist eine Splinekurve der  $D_{\text{Li+K}}$  (vgl. Fußnote 12) Daten.

gen Kalium ein fast konstant exponentielles Verhalten aufweist.<sup>12</sup> Damit ist klar gezeigt, daß das Abfallen der Diffusionskonstanten in ternären Systemen mit dem verwendeten Modellpotential nicht nur ein Konzentrationseffekt im eben beschriebenen Sinne ist.

Man sieht aber auch noch, daß die die Diffusionskonstante für Li in LS2 niedriger ist, als die für K in KS2. Dies stimmt qualitativ mit experimentell bestimmten Diffusionskonstanten überein, welche von niedrigeren Temperaturen zur Temperatur 1000 K extrapoliert wurden (vgl. Abbildung 5.15). Wie schon mehrfach betont wurde, muß man dabei bedenken, daß die Diffusionsdaten, die für Temperaturen  $T < T_g^{\text{sim}} \approx 1600$  K aus der Simulation bestimmt wurden, keine Gleichgewichtsdaten sind. Daher kann es auch reiner Zufall sein, daß das qualitative Verhalten der Diffusionskonstanten aus den binären Systemen dem entspricht, was man im Experiment beobachtet.

<sup>12</sup>Die gestrichelte Linie in Abbildung 5.19 stellt eine Interpolation (Splinekurve) für die kreuzkorrelationsfreie Diffusion  $D_{\text{Li+K}} = (1 - x) \cdot D_{\text{Li}} + x \cdot D_{\text{K}}$  dar, mit  $x = \text{K}/(\text{Li} + \text{K})$ . Sie soll eine Andeutung des Verlaufs der Gesamtdiffusion für das Auge sein.

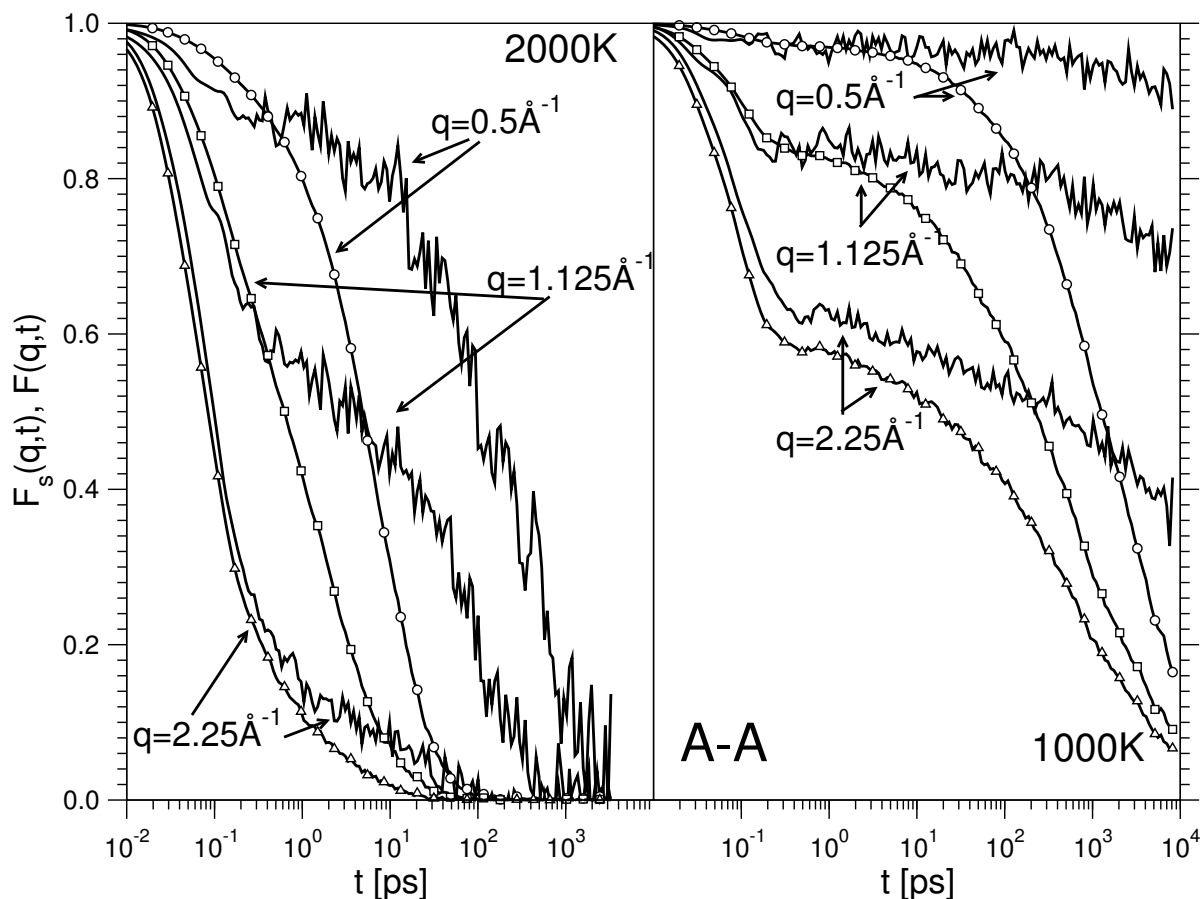
### 5.2.2. Dynamik von ternärem $0.5(\text{Li}_2\text{O}) 0.5(\text{K}_2\text{O}) 2(\text{SiO}_2)$

Nun, da bekannt ist, daß das verwendete Modell geeignet ist, den MAE zu reproduzieren, soll dieser nun mikroskopisch untersucht werden. Dabei wird im wesentlichen nach Unterschieden zu den binären Systemen gesucht, um festzustellen, warum die Alkaliionen in LKS2 vergleichsweise langsam sind. Im weiteren Verlauf der Besprechung wird oftmals nur der Vergleich der Größen für Kalium in KS2 und LKS2 durchgeführt, da die Unterschiede in den Größen deutlicher sind, als die von Lithium in LS2 und LKS2.

Die Art, wie die Alkaliionen sich in den Subnetzwerken fortbewegen, wurde schon kurz in Kapitel 5.2.1 angesprochen. Dort findet man in Abbildung 5.17 sowohl die  $G_s(r, t)$  für Kalium bei 2000 K als auch die bei 1000 K. Vergleicht man zuerst die Verteilungen für  $G_s(r, t)$  für einzelne Zeiten bei 2000 K, so stellt man fest, daß zwischen LKS2 und KS2 kein wesentlicher Unterschied in der zeitlichen Entwicklung von  $G_s(r, t)$  zu erkennen ist. Dies ist aber auch eine Temperatur, bei welcher man nur einen recht kleinen MAE beobachtet.

Anders ist das Bild bei 1000 K: Zunächst sieht man die gleichen Merkmale für K in LKS2 wie für K in KS2, nämlich, daß der Peak, der für kurze Zeiten existierte, langsam zerfällt, wobei sich dann im Bereich der NN-Distanzen stattdessen Peaks ausbilden. Soweit stimmt das Verhalten in binären und ternären System qualitativ überein, und bestätigt, daß auch im ternären System sich die Ionen durch Sprünge (Hopping) auf Nachbarplätze fortbewegen. Betrachtet man aber jetzt genauer, wann diese Verteilungen  $G_s(r, t)$  eine gewisse Form annehmen, oder Peaks an bestimmten NN-Distanzen ausbilden, so erkennt man, daß in LKS2 zwar die gleichen Verteilungen  $G_s(r)$  auftreten, allerdings auf einer Zeitskala, die ca. eine Größenordnung langsamer ist als in KS2. Das erkennt man z.B. an den  $G_s(r, t)$  für  $t = 40$  ps in KS2 und  $t = 398$  ps in LKS2. Bei  $G_s(r, t)$  für  $t = 40$  ps in KS2 erkennt man, daß sich auf der ersten NN-Distanz ein deutlicher Peak aufgebaut hat, der eine breite Flanke bis hin zu ca.  $10 \text{ \AA}$  aufweist. Eine entsprechende Verteilung findet man auch für LKS2, allerdings handelt es sich dort um  $G_s(r, t)$  bei  $t = 398$  ps. Also war hier ein Faktor 10 in der Zeit nötig, damit  $G_s(r, t)$  in LKS2 ebensoweit zerfallen ist, wie in KS2 bei  $t = 40$  ps, was aber ungefähr die gleiche Größenordnung ist, um die die Diffusionskonstante in LKS2 im Vergleich zu KS2 abgefallen ist ( $\propto 8.6$ ).

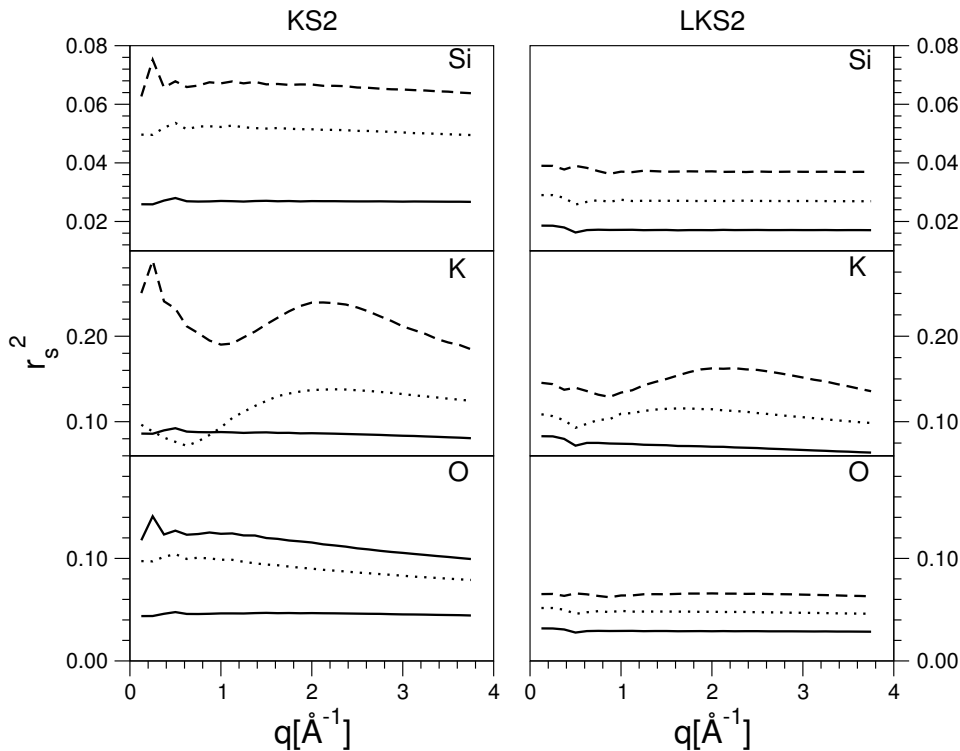
Um weitere Vergleiche mit den binären Systemen vorzunehmen, wurden die kohärenten und inkohärenten Streufunktionen in Abbildung 5.20 analog zu Abbildung 3.18 berechnet. Dabei wurde diesmal nicht nur eine Teilchensorte alleine untersucht, sondern die beiden Alkaliionensorten wurden zusammen in die Berechnung mit einbezogen ( $A = \{\text{Li}, \text{K}\}$ ). Genau wie bei den binären Systemen beobachtet man schon bei 2000 K eine Zeitskalentrennung zwischen dem Zerfall ausgezeichneter Kaliumplätze ( $F(q, t)$ ) und der schnelleren Dekorrelation der Einteilchengröße  $F_s(q, t)$ . Vergleicht man nun auch die Streufunktionen für 1000 K mit denen, die für die binären Systeme berechnet wurden (Abbildung 3.18), so stimmt auch hier das qualitative Verhalten insoweit überein, daß bei dieser Temperatur die  $F_s(q, t)$  noch zerfallen, die  $F(q, t)$  jedoch auf einem Plateauwert einfrieren. Nimmt man die Zeiten, zu denen die  $F_s(q, t)$  auf 0.1 abfallen, so erkennt man, daß die  $F_s(q, t)$  in LKS2 bei 1000 K ca. eine Größenordnung in der Zeit langsamer zerfallen, als es in KS2 oder LS2 der Fall ist. Im Vergleich auch sehr auffällig ist die etwas ausgeprägtere Abwärtsdrift des Plateauwerts der  $F(q, t)$  in LKS2. Dieser Effekt wurde in Kapitel 3.3 als Alterung erkannt, die im Falle von LKS2 deutlicher in den Streufunktionen zu erkennen ist, als bei den binären Systemen. Um das quantitativ besser



**Abbildung 5.20.:** Gemeinsame kohärente– (durchgezogene Linien) und inkohärente (Linien mit offenen Kreisen) intermediäre Streufunktionen für 2000 K (linke Abbildung) und 1000 K (rechte Abbildung), für Alkaliionen (A) in LKS2.

zu verifizieren, werden noch einmal die Resultate zu den Relaxationszeiten  $\tau_s$  aus Abbildung 3.23 betrachtet. Speziell bei Kalium erkennt man, daß die Veränderung des Verlaufs der Relaxationszeiten um 1000 K größer in LKS2 ausfällt, als in KS2. Der Knick beim Übergang vom konkaven Verlauf hin zu dem Verhalten bei niedrigen Temperaturen ist deutlich ausgeprägter. Das wiederum bedeutet, daß der Einfluß der Matrix, die sich nicht mehr im Gleichgewicht befindet, in LKS2 auf die Alkaliionen eine stärkere Änderung der Diffusion hervorruft, als in den binären Systemen.

Dieses Verhalten kann man sich möglicherweise durch die Betrachtung der  $q$ -abhängigen Lokalisierungslängen  $r_s(q)$  erklären. Es sollen nun zunächst mal die  $r_s$  aus Abbildung 5.21 besprochen werden. Das linke Bild der Abbildung zeigt noch einmal die  $r_s(q)$  von KS2, die schon in Abbildung 3.20 zu sehen waren. Verglichen werden sie mit den  $r_s(q)$  von Kalium in LKS2. Betrachtet man zunächst nur die Matrixkomponenten Si und O, so kann man darüber zwei Aussagen machen: Zum einen zeigen die  $r_s(q)$  weniger Anharmonizität, d.h. eine geringere Abhängigkeit von  $q$ , als es speziell für O in KS2 der Fall ist, wo für Temperaturen  $T > 700$  K ein Abfallen mit  $q$  zu beobachten ist. Desweiteren sind die Lokalisierungslängen für die Matrixkomponenten in LKS2 für alle Temperaturen kleiner, vor allem aber für die hö-

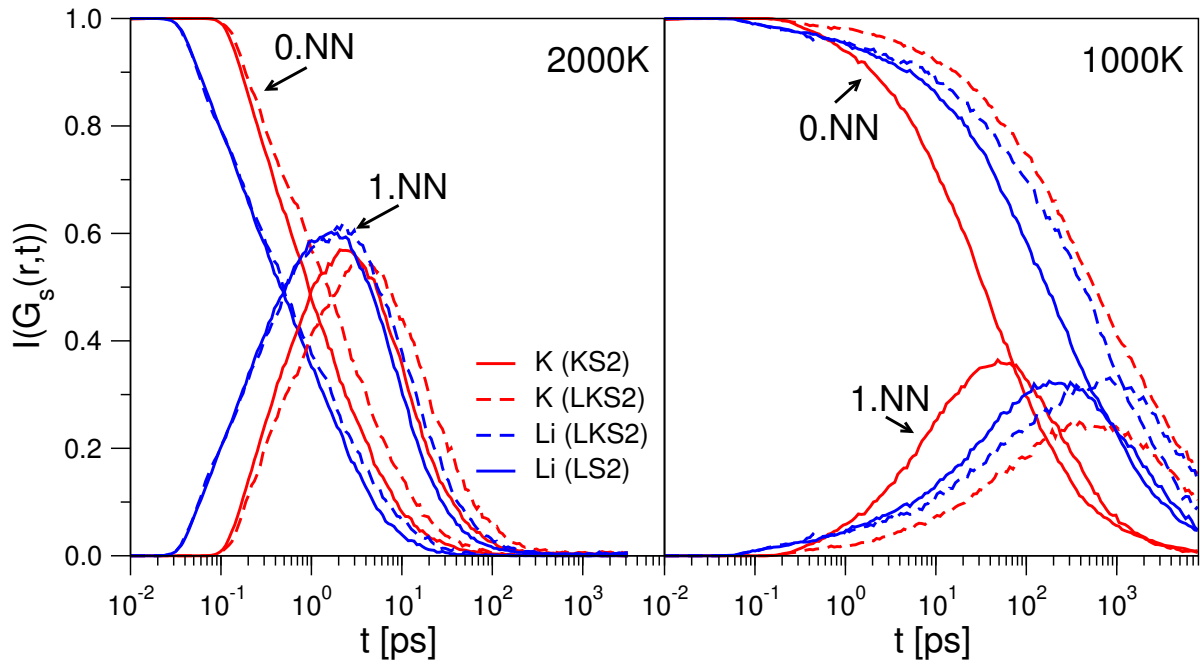


**Abbildung 5.21.:** Quadrat der Lokalisierungslängen aus den Lamb-Mößbauer-Faktoren  $r_s^2$  für die Temperaturen 700 K (durchgezogene Linie), 1000 K (gepunktete Linie) und 1250 K (gestrichelte Linie), analog zu Abbildung 3.20.

heren Temperaturen fällt der Unterschied deutlicher aus. Bei Kalium sieht das Bild ähnlich aus, wie für die Matrixkomponenten. Die Anharmonizitäten fallen in LKS2 gedämpfter, also weniger ausgeprägt aus, als in KS2. Auch werden die Lokalisierungslängen kleiner, allerdings fällt dieser Effekt nicht so deutlich aus, wie der für die Matrixkomponenten. Um zurückzukommen auf die Frage, warum der Einfluß der aus dem Gleichgewicht gefallenen Matrix auf die Alkaliionen in LKS2 stärker zu sein scheint, kann man eine Verbindung mit den Lokalisierungslängen herstellen. Da diese für alle Komponenten im System kleiner ausfällt, scheint die Anwesenheit der zweiten Alkaliionensorte im System dazu zu führen, daß das gesamte System insgesamt „steifer“ wird, also daß die Teilchen sich in deutlicher definierten Plätzen befinden, als in den binären Systemen. Es ist also so, als ob zwei Ionensorten die Matrix stützen würden, die Matrix also steifer machen. Die Matrix wiederum hat einen Einfluß auf den Transportmechanismus der Alkaliionen, wie man in den Relaxationszeiten  $\tau_s$  sehen kann. Ist die Matrix steifer, so ist die Kopplung mit der Lokalisierung der Alkaliionen stärker, was den Effekt hat, daß sich das Nichtgleichgewicht der Matrix stärker auf die Transportmechanismen der Alkaliionen auswirkt.

Eine oftmals favorisierte Theorie des MAE ist, daß die Subnetzwerke durch den Sprung andersnamiger Ionen in sie hinein „verstopft werden“.<sup>13</sup> Um die lokalen Bewegungen der Alkaliionen auf Längenskalen der ersten NN-Schale zu untersuchen, werden die „distinct“-Anteile

<sup>13</sup>vgl. dazu Kapitel 1.3.



**Abbildung 5.22.:**  $G_s(r, t)$  integriert über den Bereich der Korrelationslücke von  $g_{\alpha\alpha}(r)$  (0.NN) und über den ersten NN-Bereich für 2000 K (links) und 1000 K (rechts).

$G^{\alpha\beta}(r, t)$  und die „self“-Anteile  $G_s^\alpha(r, t)$  der van-Hove-Korrelationsfunktion herangezogen. Dazu wurde zu verschiedenen Zeiten in der Simulation die Verteilung  $G_s^\alpha(r)$  und  $G^{\alpha\beta}(r)$  aufgenommen und über bestimmte Bereiche integriert, einmal von  $r = 0$  bis zum Ende der Korrelationslücke (also die „0.“ NN-Schale) und vom Ende der Korrelationslücke bis zum ersten Minimum von  $g_{\alpha\beta}(r)$  (die erste NN-Schale).

Führt man diese Integrationen für  $G_s^\alpha(r)$  über die 0. NN-Schale aus ( $I(G_s^\alpha(r)|_0)$ ), so lernt man etwas über die Anzahl und die Verweildauer der Teilchen an ihren Orten zur Zeit  $t = 0$ . Bewegen sich die Teilchen aus den Bereich der 0. NN-Schale heraus, populieren sie den Bereich der ersten NN-Schale. Durch das Integral von  $G_s^\alpha(r)$  über die erste NN-Schale ( $I(G_s^\alpha(r)|_1)$ ) erfährt man wiederum etwas über Anteil und Verweildauer der Teilchen in der ersten NN-Schale. Diese Integrale wurden in Abbildung 5.22 für eine hohe (2000 K) und eine niedrige Temperatur (1000 K) aufgetragen.

Es soll zunächst die hohe Temperatur genauer betrachtet werden. In der 0. NN-Schale erkennt man für die Alkaliionen in binären Systemen und die in ternären Systemen einen im wesentlichen gleich schnellen Zerfall, der in intermediären Zeiten (ca.  $10^0 - 10^1$  ps) in LKS2 etwas verzögert abläuft. Generell zerfallen die  $I(G_s^\alpha(r, t)|_0)$  aber in der Zeitgrößenordnung 10 ps auf 0.1, also ähnlich schnell wie die  $F_s(q, t)$  (Abbildung 5.20). Analog das Bild für  $I(G_s^\alpha(r, t)|_1)$ . Parallel zum Zerfall von  $I(G_s^\alpha(r, t)|_0)$  sich ein Peak in  $I(G_s^\alpha(r, t)|_1)$  auf, der sein (zeitabhängiges) Maximum ungefähr dann erreicht, wenn  $I(G_s^\alpha(r, t)|_0)$  seinen Wendepunkt durchläuft. Danach bewegen sich die Teilchen auch weg von der ersten NN-Schale, weshalb der Peak sich abbaut. Die Vorgänge für  $I(G_s^\alpha(r, t)|_1)$  sind hier auch in der gleichen zeitlichen Größenordnung für die binären Systeme und LKS2.

Jetzt soll die niedrige Temperatur betrachtet werden. Auffällig ist sofort, daß die Prozesse

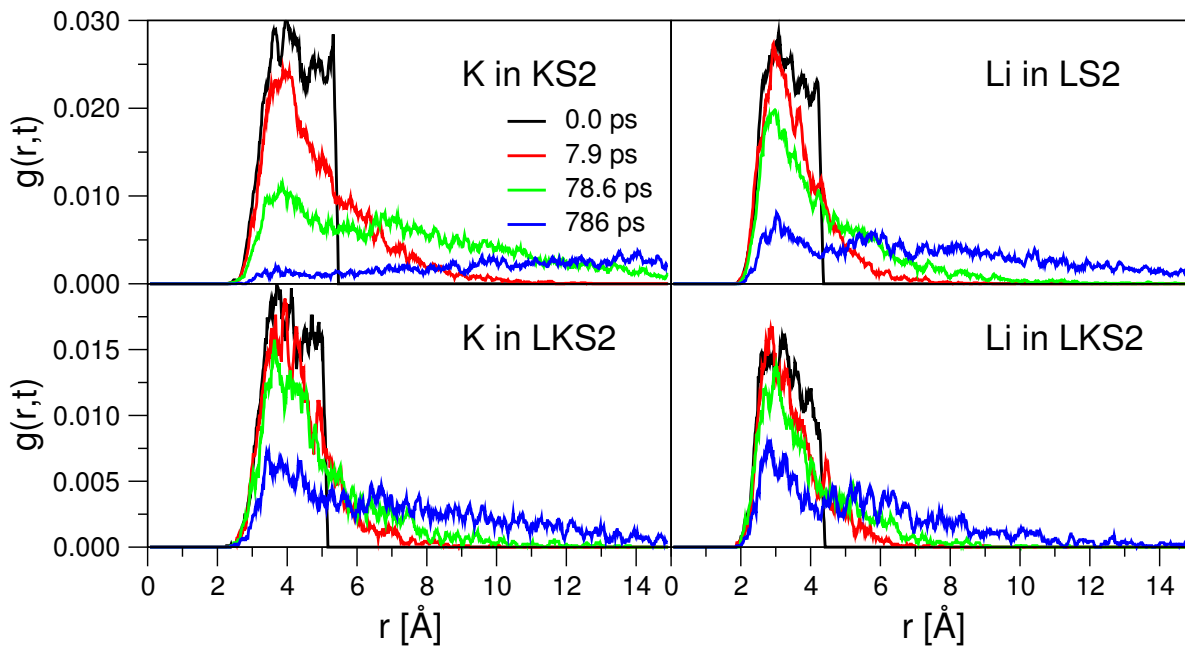


alle deutlich langsamer geworden sind. Der Zerfall von  $I(G_s^\alpha(r, t)|_0)$  auf 0.1 braucht jetzt  $10^3 - 10^4$  ps, die Teilchen bleiben also ca. 3 Größenordnungen länger mit ihrer Ausgangsposition korreliert (was nicht heißt, daß sie nicht zwischendurch weg- und zurückspringen können). Desweiteren erkennt man nun zwischen den binären Systemen und LKS2 eine deutliche Trennung der Dekorrelationszeiten, speziell bei Kalium erkennt man zwischen KS2 und LKS2 einen Zeitunterschied von einer Größenordnung, was wieder einmal die Zeitverzögerung ist, die bisher immer wieder in allen dynamischen Größen für K in KS2 und LKS2 aufgetreten ist.

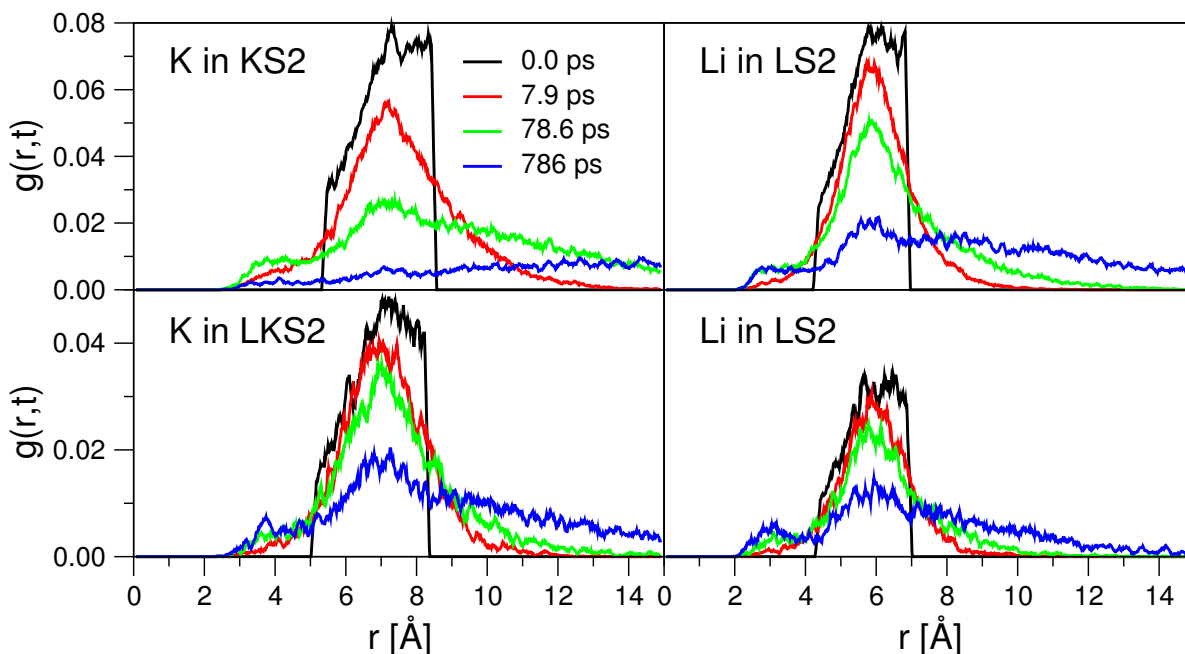
Es gibt aber auch weitere Korrelatoren, in denen man diese charakteristische Zeitverzögerung sieht. Beispiele dafür finden sich in Abbildung 5.23 und in Abbildung 5.24. Darin sieht man die zeitliche Entwicklung der Abstandsverteilung von Teilchen, die zum Zeitpunkt  $t = 0$  markiert wurden. Kriterium für das Markieren ist dabei, ob sich die Teilchen in der ersten NN-Schale (Abbildung 5.23) oder in der zweiten NN-Schale (Abbildung 5.24) befanden. Die Temperatur der Systeme, aus denen die Verteilungen stammen, beträgt 1000 K. Mittels dieser Auswertung sollte überprüft werden, ob langzeitige Korrelationen zwischen den Teilchen mit den um sie liegenden Teilchen zu erkennen sind. Sowohl im Fall der Teilchen in der ersten als auch der zweiten NN-Schale sind allerdings keine langanhaltenden Korrelationen zu erkennen. Kommen die Systeme in die Zeitgrößenordnung der Diffusion, verschmieren die Verteilungen ohne erkennbare Struktur. Im Falle der zweiten NN-Schale sieht man im Bereich der ersten NN-Schale zwar für mittlere Zeiten einen Peak, der aber auch mit zunehmender Simulationszeitdauer verwäscht. Beispiel: K in KS2. Dort zerfällt die zweite NN-Schale sowohl hin zu größeren  $r$  als auch hin zu kleineren  $r$ . Die Teilchen, die sich zunächst zu kleineren  $r$  hin bewegen, bilden einen schwach ausgeprägten Peak im Abstand der ersten NN, dieser erreicht sein Maximum aber schon nach ca. 80 ps. Simuliert man weiter, erkennt man eine Zeitgrößenordnung später keinerlei Struktur mehr in diesen Distanzen. Aber in beiden Abbildungen sieht man folgendes: Vergleicht man für Kalium die zeitabhängigen Verteilungen in KS2 und LKS2, so erkennt man, daß sie sich gleichen, wenn man die Verteilungen in KS2 mit denen vergleicht, die in LKS2 eine Zeitgrößenordnung später aufgenommen wurden, z.B. die Verteilungen zur Zeit 78.6 ps in KS2 mit denen in LKS2 zur Zeit 786 ps.

Dessen ungeachtet gibt es noch eine weitere Größe, in der diese Verzögerung um eine Zeitgrößenordnung zu beobachten ist. Dazu wurden die Winkelverteilungen von Alkaliionenpaaren aufgenommen, die zum Zeitpunkt  $t = 0$  NN waren (für Kalium in KS2 und LKS2 in Abbildung 5.25). Es ist nicht weiter verwunderlich, daß diese Korrelationen mit dem Einsetzen der diffusiven Bewegung zerfallen. Das liegt daran, daß die Netzwerke nicht geradlinig durch die Matrix führen, sondern verwinkelt (wie man anhand der Bondwinkelverteilungen sehen kann). Was man aber aus Abbildung 5.25 erkennen kann, ist, daß der Zerfall der Winkelkorrelationen hier wiederum in LKS2 eine Größenordnung verzögert zum Zerfall in KS2 stattfindet (vgl. dazu die Winkelverteilungen zur Zeit 78.6 ps in KS2 und 786 ps in LKS2).

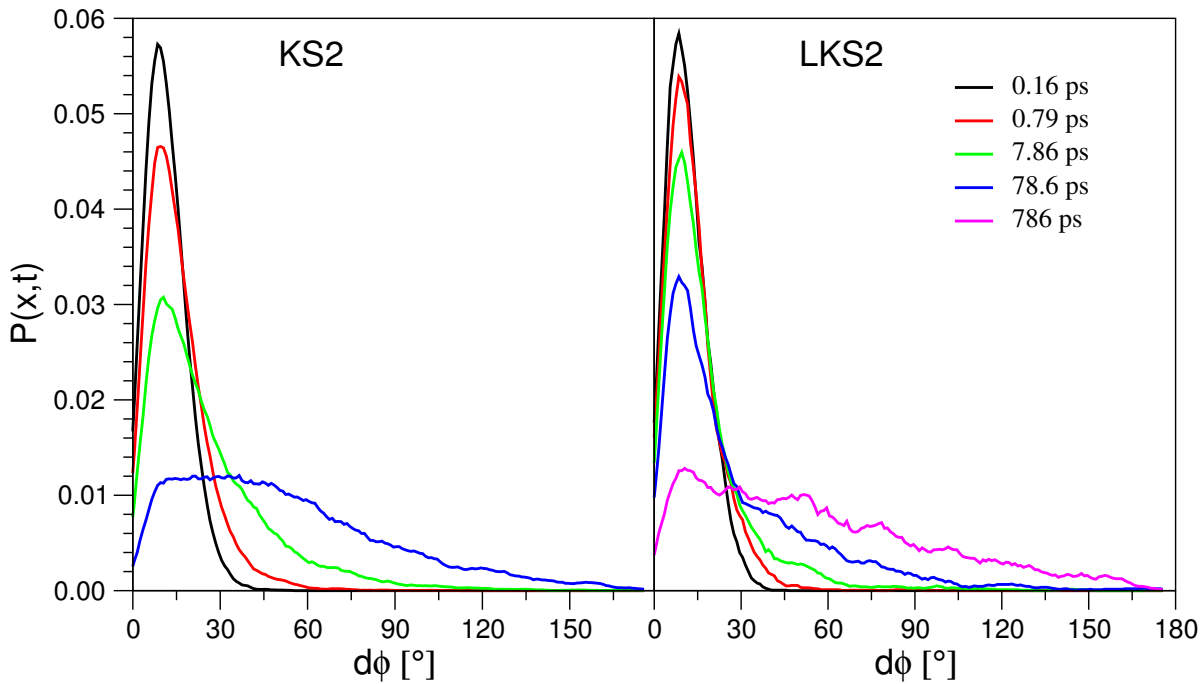
Es stellt sich die Frage, ob diese in Kalium beobachteten Verzögerung in LKS2 einfach daher rührt, daß die Teilchen länger brauchen, um von ihrer Ausgangsposition wegzuspringen, bzw., ob einfach weniger Teilchen pro Zeiteinheit springen. Um diese Frage zu beantworten, wurden nach dem Verfahren das in Kapitel 3.3 erläutert wurde, die SQD's der Alkaliionen aufgenommen, und der Zeitpunkt bestimmt, an dem ein Teilchen zum ersten Mal von dem Platz, den es zum Zeitpunkt  $t = 0$  der Simulation innehatte, wegspringt. Diese Wahrscheinlichkeit wird hier  $P_{\text{FJT}}(t, t_0)$  genannt (FJT=First Jump Time). Die Ergebnisse für 1000 K sieht man in Abbildung 5.26. Dort ist die Anzahlverteilung der Sprünge nach einer Verweildauer  $t$  auf-



**Abbildung 5.23.:** Paarkorrelation zu verschiedenen Zeiten von vorselektierten Teilchen bei 1000 K. Markiert wurden zur Zeit  $t = 0$  diejenigen gleichnamigen Teilchen, die sich in der ersten NN-Schale befanden ( $\{X\}$ ). Deren Trajektorien wurden dann weiterhin verfolgt um ein  $g(r_{\{X\}}, t)$  zu bestimmen (Darstellung über 10 Punkte geglättet).



**Abbildung 5.24.:** Paarkorrelation analog zu Abbildung 5.23, hier wurden allerdings die Teilchen der zweiten Schale markiert.

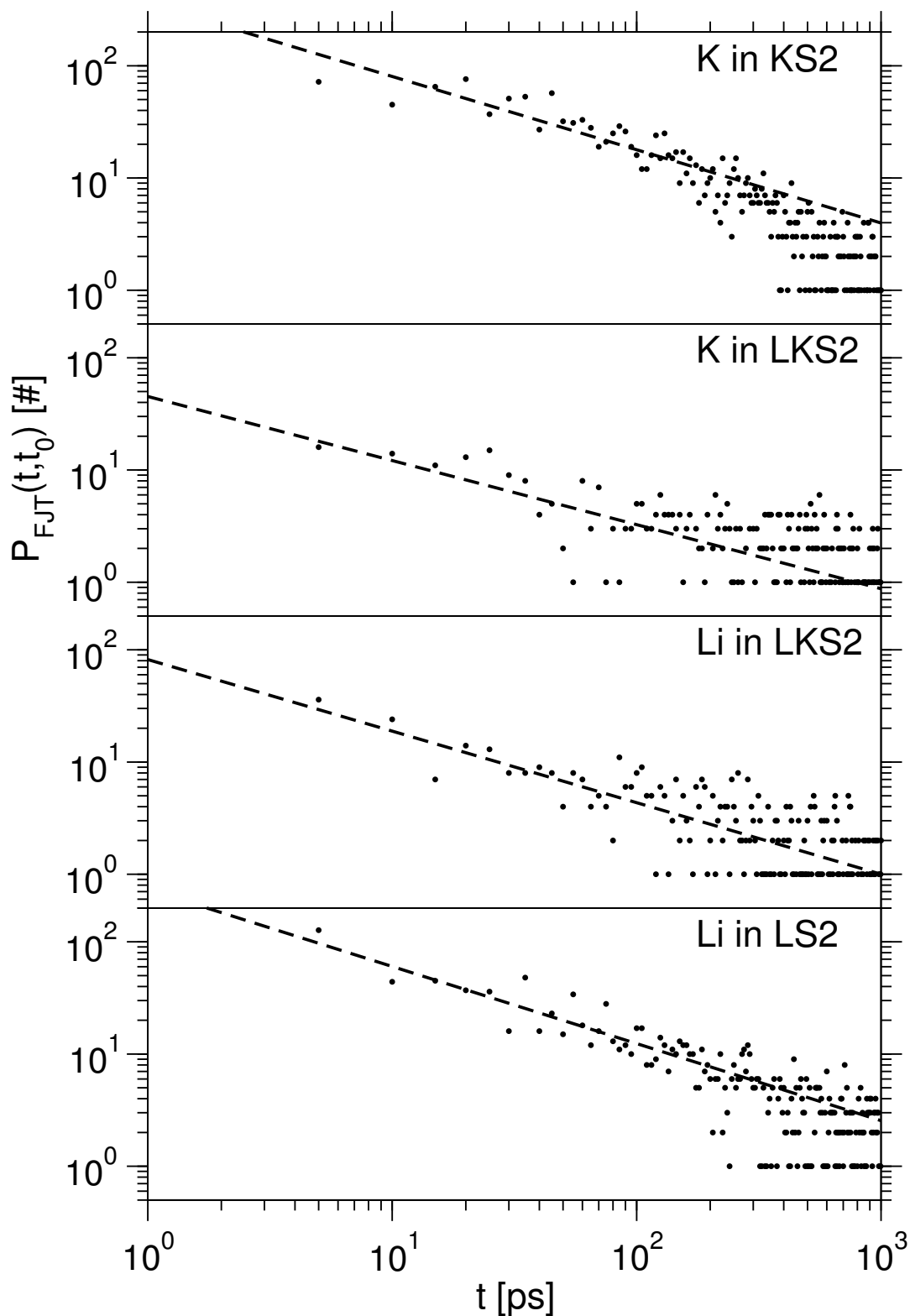


**Abbildung 5.25.:** Wahrscheinlichkeit  $P(x,t)$ , daß sich nach einer Zeit  $t$  die Richtung des Verbindungsvektor zweier Alkaliionen (hier Kalium), die zur Zeit  $t = 0$  NN waren, um  $d\phi$  verändert hat (Darstellung über 10 Punkte geglättet).

getragen. Wie in Abbildung 5.26 mittels gestrichelter Linie angedeutet, lassen sich die Daten gut mittels eines Potenzgesetzes fitten. Addiert man die Ergebnisse für Li und K in LKS2 zusammen, so erhält man in Anzahl und Zeit ähnliche Werte, wie für die binären Systeme. Zum Vergleich: In den in Abbildung 5.26 geplotteten zeitabhängigen Sprungwahrscheinlichkeiten springen im aufgetragenen Zeitraum in KS2 1724 Kaliumionen, in LS2 1597 Lithiumionen und in LKS2 1414 Alkaliionen. Also kann die langsamere Dynamik der Alkaliionen in LKS2 nicht auf verringerte oder verzögerte Sprünge zurückgeführt werden.

Es soll hier die Betrachtung der  $I(G_s^\alpha(r,t)|_0)$  und  $I(G_s^\alpha(r,t)|_1)$  fortgeführt werden. Die  $I(G_s^\alpha(r,t)|_1)$  zeigen ein entsprechend verzögertes Verhalten wie die  $I(G_s^\alpha(r,t)|_0)$ , d.h., die Peakpositionen sind um ca. 3 Größenordnungen zu späteren Zeiten hin verschoben. Zudem sind die Verteilungen breiter und die Peakhöhen geringer. Konzentriert man sich auf die Relation der Verläufe zwischen den  $I(G_s^\alpha(r,t)|_0)$  und den  $I(G_s^\alpha(r,t)|_1)$ , ist unübersehbar, daß bei der hohen Temperatur die  $I(G_s^\alpha(r,t)|_1)$  erst nach den  $I(G_s^\alpha(r,t)|_0)$  zerfallen, bei der niedrigen Temperatur jedoch die Integrale für längere Zeiten gemeinsam abfallen.

Was ist der Grund für dieses Verhalten? Um dieses Verhalten besser zu verstehen, wird hier auf ein einfaches Kompartimentmodell zurückgegriffen [135]. Schematisch ist das Modell in Abbildung 5.27 dargestellt. Das Kompartiment  $X$  repräsentiert dabei die Gesamtheit der Teilchen im Bereich der 0. NN-Schale,  $Y$  die Gesamtheit im Bereich der ersten NN-Schale.  $\alpha$  ist dabei der Anteil an Teilchen, die pro Zeiteinheit nach  $Y$  abfließen,  $\beta$  gibt den Anteil der Rücksprünge aus  $Y$  an und  $\gamma$  ist der Koeffizient der Teilchen, der in Schalen jenseits der ersten NN springt. Es handelt sich hierbei um ein stark vereinfachtes Modell, in dem



**Abbildung 5.26.:** Nicht normierte Wahrscheinlichkeit, daß ein Teilchen ausgehend von einer Konfiguration  $\Gamma(t_0)$  auf einen Nachbarplatz springt ( $P_{\text{FJT}}(t, t_0)$ ). Hier exemplarisch zu sehen ist  $P_{\text{FJT}}(t, t_0)$  für Li und K in LS2, LKS2 und KS2 bei 1000 K. Die gestrichelte schwarze Linie stellt einen Power-Law-Fit an die Daten für  $t < 200$  ps dar.

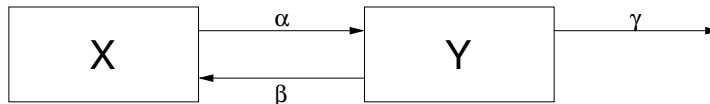


Abbildung 5.27.: Schema zum Kompartimentmodell.

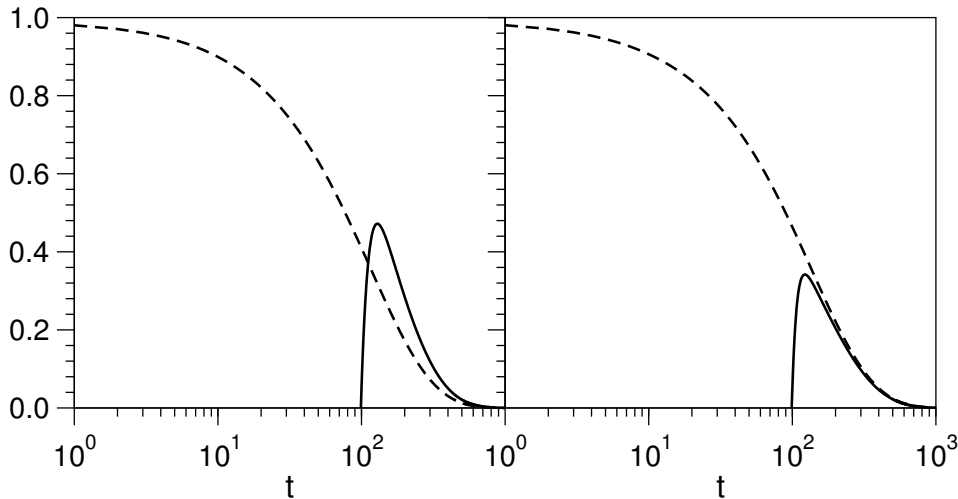
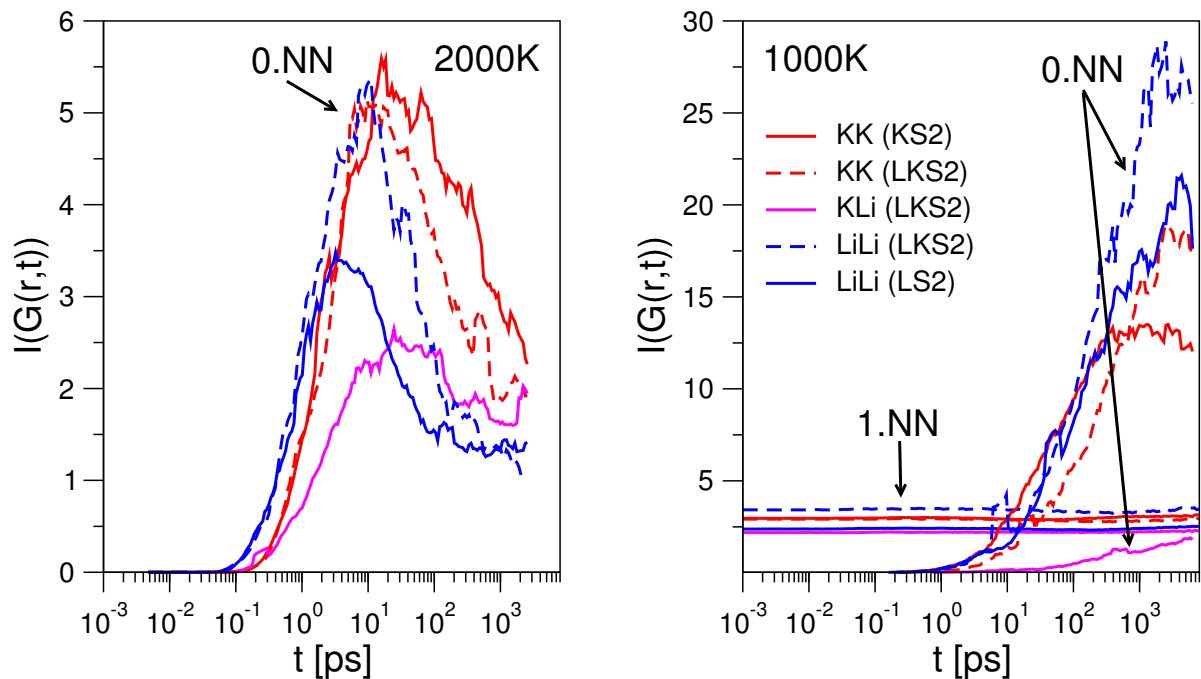


Abbildung 5.28.: Illustration zum Kompartimentmodell. Die  $y$ -Achse stellt eine normierte Besetzungszahl der Kompartimente dar und die Zeitskala ist beliebig. Die gestrichelte Linie ist die Besetzung des Kompartiments  $X$ , die durchgezogene die von  $Y$  (wobei diese in  $y$ -Richtung mit dem Faktor 5 gestreckt wurde, um sie besser zu sehen).

einige vereinfachende Annahmen gemacht werden. Einerseits werden z.B. keine Rücksprünge nach  $Y$  von Teilchen, die über  $\gamma$  abgefließen sind, modelliert. Diese Vereinfachung ist nötig, um nicht  $n \gg 1$  Kompartimente modellieren zu müssen. Zum anderen werden hier keine Wartezeiten berücksichtigt, der Teilchenstrom fließt im Rahmen der Diskretisierung instantan. Das dies nicht richtig ist, kann man sich anhand eines einfachen Beispiels überlegen: Springt ein Teilchen aus  $X$  nach  $Y$ , so würde es schon beim nächsten Zeitschritt teilweise (aufgrund der kontinuierlichen Formel) zurück nach  $X$  fließen. Dies kann aber nicht richtig sein, da jedes Teilchen eine mittlere Verweildauer auf einem Platz aufweist (siehe z.B. Abbildung 5.26). Daher wurden zur Zeitachse von  $Y$  eine mittlere Wartezeit  $\omega$  (hier wurde  $\omega = 80$  gewählt) hinzuaddiert, um die Verweildauer vereinfacht zu berücksichtigen. Konkret wird angesetzt, daß

$$\begin{aligned}
 x(t) &= x(t-1) + \beta \cdot y(t'-1) - \alpha \cdot x(t-1) \quad \text{und} \\
 y(t') &= y(t'-1) + \alpha \cdot x(t-1) - (\beta + \gamma) \cdot y(t'-1) \quad \text{mit } t' = t + \omega,
 \end{aligned}$$

mit  $x(t)$  der Besetzungszahl von  $X$  und  $y(t)$  der Besetzungszahl von  $Y$ . Um die Parameter  $\beta$  und  $\gamma$  zu bestimmen, müssen noch einige Modellannahmen gemacht werden. Zum einen wird angenommen, daß jeder Teilchenplatz identisch in der Rate des Abflusses ist, d.h.  $\alpha$  ist die Abflußrate aus sowohl  $X$  als auch  $Y$ . Aber, es wird angenommen, daß jedes Kompartiment 2



**Abbildung 5.29.:**  $G^{\alpha\beta}(r, t)$  integriert über den Bereich der Korrelationslücke von  $g_{\alpha\beta}(r)$  (0.NN) und über den ersten NN-Bereich für 2000 K (links) und 1000 K (rechts).

Nachbarkompartimente hat, also repräsentiert  $Y$  zwei Kompartimente in einer eindimensionalen Kompartimentkette, so daß es zwischen den zwei Nachbarkompartimenten von  $X$  untereinander keinen Teilchenaustausch gibt. Ist jetzt die Wahrscheinlichkeit für einen Wegsprung und einem Rücksprung immer die gleiche, so ergibt sich  $\beta = \gamma = 2 \cdot \alpha$ . Die resultierende normierte Besetzungszahl ist in Abbildung 5.28 links zu sehen. Man erkennt ein ähnliches Verhalten wie in Abbildung 5.22 für die hohe Temperatur. Nimmt man aber jetzt an, daß Rücksprünge von  $Y$  nach  $X$  doppelt so wahrscheinlich sind, also  $\beta = 4 \cdot \alpha$  und  $\gamma = 2 \cdot \alpha$ , so erhält man ein Ergebnis, daß qualitativ mit dem Verhalten in Abbildung 5.22 für niedrige Temperaturen übereinstimmt. Das bedeutet also, daß Rücksprünge in den vorherigen Platz für niedrige Temperaturen immer wahrscheinlicher werden. Die Auswirkung dessen wird am Ende dieses Kapitels noch im Kontext mit anderen Erkenntnissen dieser Arbeit diskutiert.

Bisher wurde nur der „self“-Anteil der van-Hove-Korrelationsfunktion besprochen. Was noch nicht betrachtet wurde, ist das Integral der „distinct“-Anteile der van-Hove-Korrelationsfunktion für die Alkaliionen. Analog zu Abbildung 5.22 wurde in Abbildung 5.29 für die gleichen Temperaturen und die gleichen Abstandsbereiche  $G^{\alpha\beta}(r)$  integriert ( $I(G^{\alpha\beta}(r, t)|_0)$  und  $I(G^{\alpha\beta}(r, t)|_1)$ ). Diese Integrale zeigen, wieviele Teilchen der Sorte  $\beta$  sich zur Zeit  $t$  sich an den Orten der Teilchen  $\alpha$  zur Zeit  $t = 0$  aufhalten. Ist  $\alpha = \beta$ , so sieht man, welcher Anteil der Teilchen in die Plätze springt, in der vorher ein gleichnamiges Teilchen war. Ist  $\alpha \neq \beta$ , so erfährt man, welcher Anteil der Teilchen den Platz getauscht hat.

Wendet man seine Aufmerksamkeit auf den linken Teil der Abbildung 5.29, so sieht man, daß für hohe Temperaturen die Alkaliionen durchaus gerne in die Plätze gleichnamiger Ionen springen, daß aber ab ca.  $10^1 - 10^2$  ps der Anteil abfällt. Das ist aber auch der Zeitbereich, in

dem man in Abbildung 5.20 erkennen kann, daß die  $F(q, t)$  für die Alkaliionen abfallen, also daß die Plätze selber keine Korrelation mehr aufweisen. Das Integral  $I(G^{\text{KLi}}(r, t)|_0)$  ist hierbei von besonderem Interesse, da es nämlich zeigt, wieviele Kaliumionen auf Lithiumplätze springen und vice versa. Man erkennt hier, daß Sprünge in ungleichnamige Plätze etwas verzögert auftreten, aber von der Amplitude von  $I(G^{\text{KLi}}(r, t)|_0)$  her ungefähr halb so häufig vorkommen, wie Sprünge in gleichnamige Plätze. Nicht weiter verwunderlich ist, daß für längere Zeiten bei der hohen Temperatur alle  $I(G^{\alpha\beta}(r, t)|_0)$  gegen den gleichen Wert streben, da die Systeme dann alle diffusiv und alle Platz–Platz–Korrelationen zerfallen sind.

Für niedrige Temperaturen ist die Frage, wie groß  $I(G^{\text{KLi}}(r, t)|_0)$  ist, von besonderem Interesse. Das liegt daran, daß einige Theorien zum MAE darauf basieren, daß fremdnamige Ionen in das Subnetzwerk einer Ionensorte springen und dieses Subnetzwerk quasi „verstopfen“. Wie man aber links in Abbildung 5.29 sehen kann, ist der Anteil der Ionen, die in fremdnamige Plätze springen, sehr klein. Dazu kommt noch, daß die  $I(G^{\text{KK}}(r, t)|_0)$  und  $I(G^{\text{LiLi}}(r, t)|_0)$  in LKS2 häufiger pro Zeiteinheit auf gleichnamige Plätze springen, als es selbst in binären Systemen der Fall ist (was man daran erkennt, daß die Amplituden von  $I(G^{\alpha\alpha}(r, t)|_0)$  in LKS2 höher sind). Bemerkenswert ist auch das Ergebnis für die  $I(G^{\alpha\beta}(r, t)|_1)$ : Sie zeigen über den untersuchten Zeitraum kaum Veränderung, was darauf zurückzuführen ist, daß die Platz–Platz–Korrelationen auf der Zeitskala der Matrix zerfallen, was im Falle der niedrigeren Temperatur nicht mehr im Simulationszeitraum passiert.

Insgesamt zeigt sich also, daß das verwendete Modellpotential in der Lage ist den MAE qualitativ zu reproduzieren. Die Bewegung durch die Subnetzwerke findet wie in den binären Systemen durch Hopping zwischen ausgezeichneten Alkaliplätzen statt. Die bei den binären Systemen beobachtete Zeitskalentrennung zwischen Alkaliionen und Matrix findet sich auch in LKS2. Alterungseffekte scheinen in LKS2 ausgeprägter zu sein, als es in den binären Systemen der Fall war. Dies wird auf eine „steifere“ Kopplung zwischen den Subnetzwerken und der Matrix zurückgeführt, die sich in stärkerer Lokalisierung von Teilchenpositionen bemerkbar macht. Zudem wurde gezeigt, daß die Zahl der Sprünge von Ionen pro Zeit sich nicht wesentlich zwischen binären und ternären Systemen unterscheidet. Verbindet man das mit den Ergebnissen aus dem Selbstanteil der Van–Hove–Korrelationsfunktion, daß sich die Alkaliionen um eine Größenordnung langsamer im System ausbreiten, könnte man vermuten, daß die Zahl der Rücksprünge in LKS2 höher ist. Es stellt sich jedoch mithilfe eines einfachen Kompartimentmodell heraus, daß die erhöhte Wahrscheinlichkeit von Rücksprüngen eine Funktion der Temperatur ist, die auch für binäre Systeme gilt. Allerdings muß man im Fall der Rücksprünge, als auch in anderen berechneten Größen beachten, daß sich das System für niedrige Temperaturen nicht im Gleichgewicht befindet. Deshalb kann man die Aussagen der Simulation nicht auf experimentelle Ergebnisse übertragen, da die experimentellen Systeme eine ganz andere thermische Vorgeschichte haben. Es werden viele dynamische Größen betrachtet, die qualitativ für binäre wie ternäre Systeme die gleichen Verteilungen aufweisen, jedoch in den ternären Systemen auf einer langsameren Zeitskala geschehen. Eine häufige Vermutung, daß die Verlangsamung der Dynamik durch „Verstopfen“ der Subnetzwerke mit fremdnamigen Alkaliionen entsteht, steht entgegen, daß nur ein sehr geringer Anteil der Ionen Sprünge auf fremdnamige Alkaliionenplätze vollführt.

## 6. Zusammenfassung

In der vorliegenden Arbeit wird mittels Molekulardynamik(MD)–Computersimulationen die Dynamik von verschiedenen Alkalisilikaten in der Schmelze und im Glas untersucht. Dazu wurde ein Paarpotential vom Gilbert–Ida–Typ mit Parametern von Tsuneyuki et al. [70] ( $\text{SiO}_2$ ) und Habasaki and Okada [71] (Alkaliionen) verwendet. Simuliert wurden Systeme mit 1152 und 8064 Teilchen im Temperaturbereich von 4000 K bis hinunter zu 700 K. Bei 1000 K wurde zudem noch ein 78.6 ns langer Simulationslauf durchgeführt, anhand dessen Alterungseffekte analysiert wurden. Gegenstand der Untersuchung waren die binären Alkalisilikate Lithiumdisilikat (LS2) und Kaliumdisilikat (KS2), das ternäre System LKS2, in welchem 16.6 mol% von sowohl Lithium– als auch Kaliumoxid enthalten ist, und ein binäres Lithiumdisilikat, in dem ein Lithiumion gegen ein Kaliumion getauscht wurde (LS2+1K).

Es ist bekannt, daß Alkalisilikate ionenleitend sind, was auf eine hohe Mobilität der Alkaliionen im Vergleich zu den glasbildenden Komponenten Si und O zurückzuführen ist. Die Simulationen von LS2 und KS2 zeigen, daß der Ursprung der hohen Mobilität der Alkaliionen mit der mikroskopischen Struktur verknüpft ist. KS2 und LS2 weisen auf intermediären Längenskalen Ordnung auf, die in partiellen statischen Struktur Faktoren durch Prepeaks reflektiert ist (für Li in LS2 liegt der Prepeak bei  $q = 1.2 \text{ \AA}^{-1}$ , bei K in KS2 gibt es Prepeaks bei  $q_{2a} = 0.5 \text{ \AA}^{-1}$  und  $q_{2b} = 0.7 \text{ \AA}^{-1}$ ). Die den Prepeaks zugrundeliegende Struktur erklärt sich durch perkolierende Netzwerke aus alkalioxidreichen Kanälen, die als Diffusionskanäle für die mobilen Alkaliionen fungieren. In diesen Kanälen bewegen sich die Ionen mittels Sprüngen (Hopping) zwischen ausgezeichneten Plätzen. Diese Plätze unterscheiden sich für die verschiedenen Ionensorten durch lokale Längenskalen (Abstände nächster Nachbarn von Alkali–Sauerstoff) und der chemischen Umgebung der Alkaliionen, die durch verschiedene mittlere Anzahlen nächster Sauerstoffnachbarn geprägt ist. In diesen Diffusionskanälen bewegen sich die Alkaliionen deutlich schneller fort, als es die Matrixkomponenten Si und O tun. Diese Zeitskalentrennung gilt aber nicht für die Plätze, auf denen sich die Alkaliionen bewegen: Anhand der kohärenten Streufunktionen kann man sehen, daß obwohl die Ionen sich schnell durch die Matrix bewegen, die Plätze, auf denen sie es tun, auf der gleichen Zeitskala zerfallen wie die Matrix selber.

In der Simulation beobachtet man für die hohen Temperaturen ( $4000 \text{ K} \geq T \geq 1500 \text{ K}$ ) eine ähnliche Aktivierungsenergie für die Diffusion der Alkaliionen wie im Experiment. Im Experiment findet allerdings unterhalb von ca. 1200 K ein Crossover in ein Arrheniusverhalten mit höherer Aktivierungsenergie statt, welches von der Simulation nicht nachvollzogen wird. Das kann mit der in der Simulation nicht im Gleichgewicht befindlichen Si–O–Matrix erklärt werden, bei der Alterungseffekte beobachtet werden.

Bis hierhin wurden Mischungen von  $\text{SiO}_2$  mit einer Alkalioxidkomponente untersucht, im Mittelpunkt des Interesses dieser Arbeit steht aber der sog. Mischkalieffekt (MAE), der in ternären Mischungen aus  $\text{SiO}_2$  mit zwei Alkalioxiden auftritt. Gegenüber Mischungen mit nur



---

einer Alkaliionensorte weisen letztere Systeme eine signifikante Verlangsamung der Alkaliionendiffusion auf.

Am stärksten ist der MAE für eine Alkalikomponente, wenn deren Konzentrationsanteil in einem ternären Mischalkalisystem gegen 0 geht. Daher wird ein LS2-System untersucht, in dem ein Li-Ion gegen ein K-Ion getauscht wird. Der Einfluß des K-Ions ist sowohl lokal in den charakteristischen Abständen zu den ersten nächsten Nachbarn (NN) zu sehen, als auch in der orts aufgelösten Koordinationszahlverteilung bis zu Längenskalen von ca. 8.5 Å. Die Untersuchung der Dynamik des eingesetzten K-Ions zeigt, daß die Sprungwahrscheinlichkeit nicht mit der Lokalisierung, einem Maß für die Bewegung eines Teilchens um seine Ruheposition, korreliert ist, aber daß eine chemische Umgebung mit wenig Li- und vielen O-NN oder vielen Li- und wenig O-NN ein Sprungereignis begünstigt.

Zuletzt wurde das ternäre Alkalisilikat  $(0.5 \cdot \text{Li}_2\text{O})(0.5 \cdot \text{Li}_2\text{O})(2 \cdot \text{SiO}_2)$  (LKS2) untersucht, dessen Struktur alle charakteristischen Längenskalen von LS2 und KS2 aufweist. Eine Subboxenanalyse, die Raumbereiche markiert, die von Alkaliionen besucht wurden, ergibt sogar neue Längenskalen, die möglicherweise Kanalelementlängen charakterisieren.

Es stellt sich eine komplexe Struktur mit zwei perkolierenden Kanalsubnetzwerken für die Diffusion der Alkaliionen ein. Die Untersuchung der Dynamik zeigt eine geringe Wahrscheinlichkeit dafür auf, daß Ionen in ein Subnetzwerk andersnamiger Ionen springen. Auch kann gezeigt werden, daß das Modellpotential den MAE reproduzieren kann, daß also die Diffusionskonstanten in LKS2 bei 700 K bis zu einer Größenordnung langsamer sind als in KS2 bzw. LS2. Der beobachtete Effekt stellt sich zudem vom funktionalen Verlauf her so dar, wie er im Experiment beobachtet wird. Es wurde festgestellt, daß trotz der zeitlichen Verzögerung in den dynamischen Größen die Anzahl der Sprünge pro Zeit nicht geringer ist und daß für niedrige Temperaturen (d.h. im Glas) Sprünge auf den Nachbarplatz mit anschließendem Rücksprung auf die vorherige Position deutlich wahrscheinlicher sind als bei hohen Temperaturen (also in der Schmelze).

Das ursprüngliche Ziel dieser Arbeit war, den MAE, so wie er im Experiment gemessen wird, auf mikroskopischer Ebene zu untersuchen. Wie sich herausstellte, ist dies aber mittels MD-Simulationen nicht möglich. Das Problem ist, daß bei ca. 1600 K die untersuchten Systeme aus dem Gleichgewicht fallen. Daß es einen großen Unterschied zu den Experimenten gibt, ergibt sich daraus, daß dort die Diffusionskonstanten bei ca. 1200 K in ein Arrheniusverhalten mit höherer Aktivierungsenergie übergehen, was von der Simulation überhaupt nicht nachvollzogen wird. Stattdessen setzt sich in der Simulation die Arrheniusfunktion für hohe Temperaturen zu niedrigen  $T$  im wesentlichen fort. Erkennen kann man das Einfrieren der Matrix allerdings in den Relaxationszeiten der Alkaliionen.

Man kann erwarten, daß die grobe Struktur der Kanal-Subnetzwerke nicht wesentlich von der thermischen Vorgeschichte abhängt. Die Existenz von Subnetzwerken ist daher eine Vorhersage aus der Simulation, die für reale Systeme relevant sein sollte. Mikroskopische Details der Dynamik bei tiefen Temperaturen, wie z.B. die Bedeutung von Rücksprüngen, können in der Simulation nicht im Gleichgewicht analysiert werden und sind daher mit Vorsicht zu genießen. Es wurde gezeigt, daß Alterungseffekte in der Si-O-Matrix großen Einfluß auf dynamische Größen haben, die den Alkaliionentransport beschreiben (z.B. auf Diffusionskonstanten). Daher ist es nicht möglich, unterhalb von ca. 1600 K die Simulation mit Experimenten zu vergleichen, in denen die Systeme eine völlig andere thermische Vorgeschichte haben. Das stellt

aber ein Problem dar, da der MAE bei niedrigeren Temperaturen beobachtet wird, im Experiment meist in einem Temperaturbereich von ca. 700 K bis hinunter zu 300 K. Man könnte argumentieren, daß man sich dann in der Simulation auf die Untersuchung des MAE in Temperaturbereichen beschränkt, in denen das System im Gleichgewicht ist. Zum einen zeigt sich der MAE aber bei solchen hohen Temperaturen wenig ausgeprägt, zum anderen stellt sich die Frage, inwiefern Prozesse, die z.B. das veränderte Arrheniusverhalten unterhalb ca. 1200 K bewirken, für den MAE wichtig sind. Das ist ein Problem, das in bisherigen Simulationsstudien zum MAE noch nicht angesprochen wurde. Ergebnisse der Simulationsmodelle wurden hier immer direkt mit dem Experimenten verglichen. Daß Simulationen aber nicht in der Lage sind, das experimentelle Verhalten der Diffusionskonstanten nachzuvollziehen, wurde dabei nie als Problem angesehen. In dieser Arbeit wurde aber gezeigt, daß Alterungsprozesse in der Si–O–Matrix für niedrige Temperaturen für das Verhalten der Diffusionskonstanten der Alkaliionen eine wichtige Rolle spielen. Das Verständnis des Arrheniusverhaltens unterhalb ca. 1200 K mit erhöhter Aktivierungsenergie und der Zusammenhang dieses Verhaltens mit der Alterung muß das Ziel von weiteren Untersuchungen zum mikroskopischen Verständnis des MAE, so wie er im Experiment beobachtet wird, sein.





```

! ----- Code Freeze -----
--- *
! - 2.00: r^2 potential as used by Anke. error free version
! *
! -> TODO: output is not yet up to date and reports r^
12 *
! - 2.01: trying to find pressure error, see 1080 version
! *
! - 2.02: found error in pressure. Values seem reasonable.
! *
! (error was "bsio" instead of "B" in BKS_pot/analys
e1) *
! - output reports r^2 function
! *
! - 2.03: changed coolrate into Kelvin
! *
! + instead of Na and Li in parfiles, Alk1 and Alk2.
! *
! + routine to write dumpfile
! *
! + time in logfile in ps
! *
! + coolrate now is subtracted from temperature (as on
e *
! intuitivly would expect from a COOLrate).
! *
! - 2.04: correction of time in [ps] in logfile (fraction wr
ong) *
! + write pressure as [GPa] into logfile
! *
! - 2.11: neilist corrected to accept odd part. nums (see ma
xbol)*
! + got rid of obsolete tneighli
! *
! + force/neilist -> neilist/force in main loop (wrong
! *
! neilist update)
! *
! - 2.12: NPT-run version
! *
! + direct calc through habasaki-param
! *
! - 2.13: corrected unscaled velloc statement in update posi
tions*
! + corrected time in logfile
! *
! - 2.14: extending potential of Na/K, so that minimum is de
eper *
! -> using sin^2n
! *
! + speedup of NPT-sim by including following code in
force*
! - force and virial-calc pulled together (gain ca.
2%) *
! - rwald coded into force-loop (gain ca. 15%)
! *
! - (f)kwald share common calc (gain ca. 5%)
! *
! - 2.15: since sin^2n somehow fails, trying -x^4+x^2-1 [-1.
.1] *
! - 2.16: previous pots failed because pot needs to be deriv
able *
! at least two times!
! *
! - new approach: sin^2 with two exponential flanks
! *
! - 2.17: 2.16 also failed in energy tests. Will tune pot by
! *
! utilizing the C-parameter
! *
! - 2.30: took over corrections made in 2.21
! *
! CORRECTION OF MAXBOL
! *
! 1. "bad gauss" eliminated by multiplying the rando
m *
! seed with myid
! *
! 2. "momentum shift" eliminated by setting the mome
ntum *
! of each particle type to 0.
! *
! 3. "Forgotten particle" corrected in maxbol. Was d
ue *
! to parrallelization!!!
! *
! - 2.31: nvelstep linearly writes vx,vy,vz. Added modulo po
s *
! to output, so now nvelstep writes
! *
! vel(i),vel(i+1),vel(i+2),pos(i),pos(i+1),pos(i+2)
! *
! - 2.32: -MPI_REAL replaced with MPI_DOUBLE, so that progra
m *
!
! runs correctly on non Cray T3E platforms.
! *
! -complex replaced with complex(DOUBLE). same reaso
n. *
! -module MPI renamed to MYMPI.
! *****
*** *
! * Docu of cooldown::
! *
! * - cooltime: time of cooldown. if ntotstep greater, an
! *
! * equilibration (NVT) run is gonna follow on
! *
! * target temp
! *
! * - coolstep: interval in steps in which cooling is done
! *
! * must be equal to tbathstep
! *
! * - coolrate: cooling/step in Kelvin different than prev.
! *
! * definition
! *
! * - tbathstep: See above. good value: 50-150
! *
! *****
*** *
!*****
*****
!-----
-|122
!
!
!
!
!-----
-|122
module typeDefs
implicit none
integer,parameter,public :: DOUBLE = kind(1D0)
integer,parameter,public :: DEFAULT = kind(1)
integer,parameter,public :: WORD = selected_int_kind(
4) ! Word-length
integer,parameter,public :: STRINGLEN = 132
! program historic string-length
end module typeDefs
module program_wide_constants ! All "parameter" definitions
included here
use typeDefs
implicit none
integer(DEFAULT),parameter,public :: &
& partdim = 8064*3, &
& nofcpus = 16, &
& partdimloc = partdim/nofcpus,& ! Beware: partd
imloc must be integer
& nchpart = partdim/3, &
& nchpartloc = partdimloc/3, &
& maxk = 17452, &
& kmax_para = 20, &
& cor1dim = 300, &
& cor2dim = 20, &
& runlength = 14040
real(DOUBLE),save,public :: pi
end module program_wide_constants
module MYMPI
use typeDefs
implicit none
include 'mpif.h'
integer :: ierr, myid, numprocs, master
integer(DEFAULT) :: nlocal, nstart, nend
end module MYMPI
module sim_commons
use typeDefs
use program_wide_constants
implicit none

```

```

real(DOUBLE), save, public :: vol, lbox, olbox, lboxd2
integer(DEFAULT), save, public :: npart

integer(DEFAULT), save, public :: napart, nbpart, ncpart, ndpart

real(DOUBLE), save, public :: hstep, ekin, epot

real(DOUBLE), save, public :: &
& rcoaasq, rcoabsq, rcoacsq, rcoadsq, &
& rcobbsq, rcobcsq, rcobdsq, &
& rcoccsq, rcocdsq, &
& rcoddsq, &
& rcoaa, rcoab, rcoac, rcoad, &
& rcobb, rcobc, rcobd, &
& rcocc, rcood, &
& rcodd, &
& decay

real(DOUBLE), save, public :: &
& rcpolyaasq, rcpolyabsq, rcpolyacsq, rcpolyadsq, &
& rcpolybbsq, rcpolybcsq, rcpolybdsq, &
& rcpolyccsq, rcpolycdsq, &
& rcpolyddsq, &
& rcpolyaa, rcpolyab, rcpolyac, rcpolyad, &
& rcpolybb, rcpolybc, rcpolybd, &
& rcpolycc, rcpolycd, &
& rcpolydd, &
& rcpolyshaa, rcpolyshab, rcpolyshac, rcpolyshad, &
& rcpolyshbb, rcpolyshbc, rcpolyshbd, &
& rcpolyshcc, rcpolyshcd, &
& rcpolyshdd &
& , rcpolybesq, rcpolybe, rcpolyshbe ! Special
K-O-interaction

real(DOUBLE), save, public :: potshaa, potshab, potshbb, potsha
c, potshbc, potshcc, potshad, potshbd, potshcd, potshdd, &
& potshbe ! Special K-O-interaction

real(DOUBLE), save, public :: tdt24
integer(DEFAULT), save, public :: carry, i24, j24, twop24
integer(DEFAULT), dimension(24), save, public :: seed

real(DOUBLE), save, public :: virial

real(DOUBLE), save, public :: inlbox

integer(WORD), dimension(800, nchpartloc), save, public :: nei
ghwald

real(DOUBLE), save, public :: al2skin
real(DOUBLE), public :: forcefitcall

end module sim_commons

module ewald

use typeDefs, only: DOUBLE, DEFAULT
use program_wide_constants
implicit none

real(DOUBLE), save, public :: alpha, vs
real(DOUBLE), save, public, dimension(maxk) :: kvec
integer(DEFAULT), save, public :: totk

end module ewald

module charge_and_mass

use typeDefs, only: DOUBLE
use program_wide_constants
implicit none

real(DOUBLE), parameter, public :: chsi = 2.4d0
real(DOUBLE), parameter, public :: chna = 0.85d0!K!0.88d0
real(DOUBLE), parameter, public :: chli = 0.87d0!Li
real(DOUBLE), public :: cho = -1.2d0
real(DOUBLE), parameter, public :: msi = 28.086d0
real(DOUBLE), parameter, public :: mna = 39.098d0 !22.99d0

real(DOUBLE), parameter, public :: mli = 6.941d0!39.098d0

real(DOUBLE), parameter, public :: mo = 15.9994d0
real(DOUBLE), dimension(nchpartloc), save, public :: ch

end module charge_and_mass

```

```

module rsum_commons

use typeDefs, only: DOUBLE, DEFAULT
implicit none

integer(DEFAULT), save, public :: rmax, rsqmax

real(DOUBLE), save, public :: twoadrpi, alpha2, malpha, al2d2, o
ned3, erkon
real(DOUBLE), save, public :: twopi, twopidlbox

real(DOUBLE), save, public :: mbeta, twobdpi, erkonbeta, sqrcw
ald, rcewald, decayew

end module rsum_commons

module ksum_commons

use typeDefs, only: DEFAULT
use program_wide_constants
implicit none

integer(DEFAULT), save, public :: kmax, ksqmax

integer(DEFAULT), dimension(maxk), save, public :: kx_field, k
y_field, kz_field

end module ksum_commons

module potpar_commons

use typeDefs, only: DOUBLE
use program_wide_constants
implicit none

real(DOUBLE), save, public :: e2n
real(DOUBLE), dimension(nchpartloc), save, public :: m
real(DOUBLE) :: asisi, asina, asili, asio, &
& bsisi, bsina, bsili, bsio, &
& csisi, csina, csili, csio, &
& anana, anali, anao, &
& bnana, bnali, bnao, &
& cnana, cnali, cnao, &
& alili, alio, &
& blili, blio, &
& clili, clio, &
& aoo, booo, coo &
& , ako, bko, cko

end module potpar_commons

!-----
-|122
!
! Main program
!
!-----
-|122
program mixed_alkali_effect

use MYMPI
use typeDefs
use program_wide_constants, only: partdim, cor1dim, cor2dim, m
axk, nchpart
use sim_commons
use charge_and_mass
use ewald
use rsum_commons
use ksum_commons
use potpar_commons
! use potential_extension

implicit none

real(DOUBLE) :: pos(partdim), posloc(partdimloc)
real(DOUBLE) :: vel(partdim), velloc(partdimloc), acc(partdi
mloc)
real(DOUBLE) :: neighposloc(partdimloc)
real(DOUBLE) :: runpos(partdim), runposloc(partdimloc)
real(DOUBLE) :: fpart1(partdimloc)

integer(DEFAULT) :: cortim(cor1dim)

integer(DEFAULT) :: i, j, k, pindex, olddat, npart3
integer(DEFAULT) :: ntotstep, ndumpstep, time, cooltim
integer(DEFAULT) :: olseed
integer(DEFAULT) :: iseed, iseedn, uni, randn(3)
integer(DEFAULT) :: nsystdump, nprintstep, tbathtstep, coolste

```



```

        print*,"Sending conffile to processes"
    endif
    call MPI_BARRIER( MPI_COMM_WORLD, ierr )
    call MPI_BCAST(olddat, 1, MPI_INTEGER,master,MPI_COMM_WORL
D, ierr)
    call MPI_BCAST(napart, 1, MPI_INTEGER,master,MPI_COMM_WORL
D, ierr)
    call MPI_BCAST(nbpart, 1, MPI_INTEGER,master,MPI_COMM_WORL
D, ierr)
    call MPI_BCAST(ncpart, 1, MPI_INTEGER,master,MPI_COMM_WORL
D, ierr)
    call MPI_BCAST(ndpart, 1, MPI_INTEGER,master,MPI_COMM_WORL
D, ierr)
    call MPI_BCAST(lbox, 1, MPI_DOUBLE,master,MPI_COMM_WORLD,
ierr)
    call MPI_BCAST(rcooa, 1, MPI_DOUBLE,master,MPI_COMM_WORLD,
ierr)
    call MPI_BCAST(rcoab, 1, MPI_DOUBLE,master,MPI_COMM_WORLD,
ierr)
    call MPI_BCAST(rcoac, 1, MPI_DOUBLE,master,MPI_COMM_WORLD,
ierr)
    call MPI_BCAST(rcobb, 1, MPI_DOUBLE,master,MPI_COMM_WORLD,
ierr)
    call MPI_BCAST(rcobc, 1, MPI_DOUBLE,master,MPI_COMM_WORLD,
ierr)
    call MPI_BCAST(rcocc, 1, MPI_DOUBLE,master,MPI_COMM_WORLD,
ierr)
    call MPI_BCAST(rcoad, 1, MPI_DOUBLE,master,MPI_COMM_WORLD,
ierr)
    call MPI_BCAST(rcobd, 1, MPI_DOUBLE,master,MPI_COMM_WORLD,
ierr)
    call MPI_BCAST(rcocd, 1, MPI_DOUBLE,master,MPI_COMM_WORLD,
ierr)
    call MPI_BCAST(rcodd, 1, MPI_DOUBLE,master,MPI_COMM_WORLD,
ierr)
    call MPI_BCAST(rcow, 1, MPI_DOUBLE,master,MPI_COMM_WORLD,
ierr)
    call MPI_BCAST(rskinw, 1, MPI_DOUBLE,master,MPI_COMM_WORL
D, ierr)
    call MPI_BCAST(temp, 1, MPI_DOUBLE,master,MPI_COMM_WORLD,
ierr)
    call MPI_BCAST(cooltim, 1, MPI_INTEGER,master,MPI_COMM_WOR
LD, ierr)
    call MPI_BCAST(pressext, 1, MPI_DOUBLE,master,MPI_COMM_WOR
LD, ierr)
    call MPI_BCAST(mpiston, 1, MPI_DOUBLE,master,MPI_COMM_WORL
D, ierr)
    call MPI_BCAST(tbathstep, 1, MPI_INTEGER,master,MPI_COMM_W
ORLD, ierr)
    call MPI_BCAST(coolstep, 1, MPI_INTEGER,master,MPI_COMM_WO
RLD, ierr)
    call MPI_BCAST(coolrate, 1, MPI_DOUBLE,master,MPI_COMM_WOR
LD, ierr)
    call MPI_BCAST(nomtoten, 1, MPI_DOUBLE,master,MPI_COMM_WOR
LD, ierr)
    call MPI_BCAST(olseed, 1, MPI_INTEGER,master,MPI_COMM_WORL
D, ierr)
    call MPI_BCAST(hstep, 1, MPI_DOUBLE,master,MPI_COMM_WORLD,
ierr)
    call MPI_BCAST(ntotstep, 1, MPI_INTEGER,master,MPI_COMM_WO
RLD, ierr)
    call MPI_BCAST(nprintstep, 1, MPI_INTEGER,master,MPI_COMM_
WORLD, ierr)
    call MPI_BCAST(ndumpstep, 1, MPI_INTEGER,master,MPI_COMM_W
ORLD, ierr)
    call MPI_BCAST(naverstep, 1, MPI_INTEGER,master,MPI_COMM_W
ORLD, ierr)
    call MPI_BCAST(ncorrelmax, 1, MPI_INTEGER,master,MPI_COMM_
WORLD, ierr)
    call MPI_BCAST(ncorrel, 1, MPI_INTEGER,master,MPI_COMM_WOR
LD, ierr)
    call MPI_BCAST(nsystdump, 1, MPI_INTEGER,master,MPI_COMM_W
ORLD, ierr)
    call MPI_BCAST(restart, 1, MPI_INTEGER,master,MPI_COMM_WOR
LD, ierr)
    call MPI_BCAST(alpha, 1, MPI_DOUBLE,master,MPI_COMM_WORLD,
ierr)
    call MPI_BCAST(rmax, 1, MPI_INTEGER,master,MPI_COMM_WORLD,
ierr)
    call MPI_BCAST(kmax, 1, MPI_INTEGER,master,MPI_COMM_WORLD,
ierr)
    call MPI_BCAST(decay, 1, MPI_DOUBLE,master,MPI_COMM_WORLD,
ierr)
    call MPI_BCAST(decayew, 1, MPI_DOUBLE,master,MPI_COMM_WORL
D, ierr)
    if(myid == 0) then
        print*,"done"
    endif

    if (myid==master) then;print*,"Checking parameters ...";en
dif
    if(rcooa > 0.5*lbox) then; print*, 'SiSi cut off too large
'; stop; endif
    if(rcoab > 0.5*lbox) then; print*, 'SiK cut off too large
'; stop; endif
    if(rcoac > 0.5*lbox) then; print*, 'SiLi cut off too large
'; stop; endif
    if(rcobb > 0.5*lbox) then; print*, 'KK cut off too large
'; stop; endif
    if(rcobc > 0.5*lbox) then; print*, 'KLi cut off too large
'; stop; endif
    if(rcocc > 0.5*lbox) then; print*, 'LiLi cut off too large
'; stop; endif
    if(rcoad > 0.5*lbox) then; print*, 'SiO cut off too large
'; stop; endif
    if(rcobd > 0.5*lbox) then; print*, 'KO cut off too large
'; stop; endif
    if(rcocd > 0.5*lbox) then; print*, 'LiO cut off too large
'; stop; endif
    if(rcodd > 0.5*lbox) then; print*, 'OO cut off too large
'; stop; endif

    ! Setup of constants:
    ! -----

    ! for the rng

    if (myid==master) then;print*,"Init of RNG ...";endif
    twop24=16777216
    intwop24=1.d0/dfloat(twop24)
    if(restart == 0) then
        i24=24
        j24=10
        carry=0
    endif
    tdtp24=2.0/float(twop24+1)

    ! for the simulation
    if (myid==master) then;print*,"Setting and calculating con
stants for the Sim ...";endif
    vol=lbox*lbox*lbox
    invol=1.d0/vol
    lboxd2=lbox/2.d0
    inlbox=1.d0/lbox

    rsqmax=rmax*rmax+2
    ksqmax=kmax*kmax+2
    npart=npart+nbpart+ncpart+ndpart
    npart3=3*npart

    al2d2=rcow*rcow
    sqrcwald=al2d2*lbox*lbox
    rcwald=rcow*lbox

    rcooasq=rcooa*rcooa; rcoabsq=rcoab*rcoab; rcoacsq=rcoac*
rcoac; rcoadsq=rcoad*rcoad
    rcobbsq=rcobb*rcobb; rcobcsq=rcobc*rcobc; rcobdsq=rcobd*
rcobd
    rcoccsq=rcocc*rcocc; rcocdsq=rcocd*rcocd
    rcoddsq=rcodd*rcodd

    al2skin=al2d2*rskinw**2
    hsqd2=hstep**2/2.d0
    hsqd3=hstep**2/3.d0
    hstepd2=hstep/2.d0

    inmpiston=1.d0/mpiston
    oned3=1.d0/3.d0
    hstepd3=hstep*oned3
    onemeps=1.d0-coolrate
    deltat=coolrate*coolstep

    call MPI_BARRIER( MPI_COMM_WORLD, ierr )

    ! Habasaki-parameters
    a(1)= 0.8688; b(1)= 0.03285; c(1)= 23.18; ! Si
    a(2)= 2.0474; b(2)= 0.17566; c(2)= 70.37; ! O
    a(3)= 1.0155; b(3)= 0.07321; c(3)= 10.87; ! Li
    a(4)= 1.0805; b(4)= 0.08461; c(4)= 0.0; ! Na
    a(5)= 1.4081; b(5)= 0.10070; c(5)= 0.0; ! K

    ! Beware: Na is K! Li is Li.
    if (myid==master) then;print*,"Calculating potential param
eters ...";endif
    if (myid==master) then;print*,"Prefactor f0=",f0;endif
    csisi = f0 * c(1)*c(1); if (myid==master) then;print*,"csi
si=",csisi;endif ! Si-Si
    csina = f0 * c(1)*c(5); if (myid==master) then;print*,"csi
na=",csina;endif ! Si-K
    csili = f0 * c(1)*c(3); if (myid==master) then;print*,"csi
li=",csili;endif ! Si-Li
    csio = f0 * c(1)*c(2); if (myid==master) then;print*,"csi
o=",csio;endif ! Si-O
    cnana = f0 * c(5)*c(5); if (myid==master) then;print*,"cna
na=",cnana;endif ! K-K
    cnali = f0 * c(5)*c(3); if (myid==master) then;print*,"cna

```



```

li=",cnali;endif ! K-Li
  cnao = f0 * c(5)*c(2); if (myid==master) then;print*, "cna
o =",cnao ;endif ! K-O
  clili = f0 * c(3)*c(3); if (myid==master) then;print*, "cli
li=",clili;endif ! Li-Li
  clio = f0 * c(3)*c(2); if (myid==master) then;print*, "cli
o =",clio ;endif ! Li-O
  coo = f0 * c(2)*c(2); if (myid==master) then;print*, "coo
=",coo ;endif ! O-O

  bsisi = 1.d0/( b(1)+b(1) ); if (myid==master) then;print*,
"bsisi=",bsisi;endif ! Si-Si
  bsina = 1.d0/( b(1)+b(5) ); if (myid==master) then;print*,
"bsina=",bsina;endif ! Si-K
  bsili = 1.d0/( b(1)+b(3) ); if (myid==master) then;print*,
"bsili=",bsili;endif ! Si-Li
  bsio = 1.d0/( b(1)+b(2) ); if (myid==master) then;print*,
"bsio =",bsio ;endif ! Si-O
  bnana = 1.d0/( b(5)+b(5) ); if (myid==master) then;print*,
"bnana=",bnana;endif ! K-K
  bnali = 1.d0/( b(5)+b(3) ); if (myid==master) then;print*,
"bnali=",bnali;endif ! K-Li
  bnao = 1.d0/( b(5)+b(2) ); if (myid==master) then;print*,
"bnao =",bnao ;endif ! K-O
  blili = 1.d0/( b(3)+b(3) ); if (myid==master) then;print*,
"blili=",blili;endif ! Li-Li
  blio = 1.d0/( b(3)+b(2) ); if (myid==master) then;print*,
"blio =",blio ;endif ! Li-O
  booo = 1.d0/( b(2)+b(2) ); if (myid==master) then;print*,
"booo =",booo ;endif ! O-O

  asisi= f0/bsisi *exp( (a(1)+a(1)) * bsisi ); if (myid==mas
ter) then;print*, "asisi=",asisi;endif ! Si-Si
  asina= f0/bsina *exp( (a(1)+a(5)) * bsina ); if (myid==mas
ter) then;print*, "asina=",asina;endif ! Si-K
  asili= f0/bsili *exp( (a(1)+a(3)) * bsili ); if (myid==mas
ter) then;print*, "asili=",asili;endif ! Si-Li
  asio = f0/bsio *exp( (a(1)+a(2)) * bsio ); if (myid==mas
ter) then;print*, "asio =",asio ;endif ! Si-O
  anana= f0/bnana *exp( (a(5)+a(5)) * bnana ); if (myid==mas
ter) then;print*, "anana=",anana;endif ! K-K
  anali= f0/bnali *exp( (a(5)+a(3)) * bnali ); if (myid==mas
ter) then;print*, "anali=",anali;endif ! K-Li
  anaoo = f0/bnao *exp( (a(5)+a(2)) * bnao ); if (myid==mas
ter) then;print*, "anaoo =",anaoo ;endif ! K-O
  alili= f0/blili *exp( (a(3)+a(3)) * blili ); if (myid==mas
ter) then;print*, "alili=",alili;endif ! Li-Li
  alio = f0/blio *exp( (a(3)+a(2)) * blio ); if (myid==mas
ter) then;print*, "alio =",alio ;endif ! Li-O
  aooo = f0/booo *exp( (a(2)+a(2)) * booo ); if (myid==mas
ter) then;print*, "aooo =",aooo ;endif ! O-O

  if (myid==master) then;print*, "Special parameters ...";en
dif

  ako=anaoo; if (myid==master) then;print*, "ako=",ako;endif
  bko=bnaoo; if (myid==master) then;print*, "bko=",bko;endif
  cko=5.d0; if (myid==master) then;print*, "cko=",cko;endif

! If a buckingham interaction is enabled, the cutoff param
eter for the interaction has to be set in the parameter
! file. Therefore a warning has been implemented here
  if (myid==master) then;print*, "Checking parameter logic ..
.";endif
  if (((asisi/=0.d0).or.(bsisi/=0.d0).or.(csisi/=0.d0)).and.
(rcoaab<=0.1d0)) then; print*, "rcoa mismatch!";stop;endif
  if (((asina/=0.d0).or.(bsina/=0.d0).or.(csina/=0.d0)).and.
(rcoab<=0.1d0)) then; print*, "rcoab mismatch!";stop;endif
  if (((asili/=0.d0).or.(bsili/=0.d0).or.(csili/=0.d0)).and.
(rcoac<=0.1d0)) then; print*, "rcoac mismatch!";stop;endif
  if (((asio /=0.d0).or.(bsio /=0.d0).or.(csio /=0.d0)).and.
(rcoad<=0.1d0)) then; print*, "rcoad mismatch!";stop;endif
  if (((anana/=0.d0).or.(bnana/=0.d0).or.(cnana/=0.d0)).and.
(rcobb<=0.1d0)) then; print*, "rcobb mismatch!";stop;endif
  if (((anali/=0.d0).or.(bnali/=0.d0).or.(cnali/=0.d0)).and.
(rcobc<=0.1d0)) then; print*, "rcobc mismatch!";stop;endif
  if (((anaoo /=0.d0).or.(bnaoo /=0.d0).or.(cnaoo /=0.d0)).and.
(rcobd<=0.1d0)) then; print*, "rcobd mismatch!";stop;endif
  if (((alili/=0.d0).or.(blili/=0.d0).or.(clili/=0.d0)).and.
(rcocc<=0.1d0)) then; print*, "rcocc mismatch!";stop;endif
  if (((alio /=0.d0).or.(blio /=0.d0).or.(clio /=0.d0)).and.
(rcocd<=0.1d0)) then; print*, "rcocd mismatch!";stop;endif
  if (((aooo /=0.d0).or.(booo /=0.d0).or.(coo /=0.d0)).and.
(rcodd<=0.1d0)) then; print*, "rcodd mismatch!";stop;endif

  nlocal=npart/numprocs
  nstart=myid*nlocal+1
  nend=nstart+nlocal-1

! Following Habasaki's argumentation, the oxygen-charge is
supposed to be variable, to ensure charge-neutrality in the
! system. This is done here.
cho=-(npart*chsi+nbpart*chna+ncpart*chli)/ndpart

```

```

if(myid.eq.master) then
  write(6,*) "Oxygen charge = ",cho
endif

do i=1,nlocal
  pindex=i+myid*nlocal
  if(pindex <= npart) then
    ch(i)=chsi
    m(i)=msi
  else
    if(pindex <= npart+nbpart) then
      ch(i)=chna
      m(i)=mna
    else
      if(pindex <= npart+nbpart+ncpart) then
        ch(i)=chli
        m(i)=mli
      else
        ch(i)=cho
        m(i)=mo
      endif
    endif
  endif
enddo

! Calculation of all potential-shifts
call MPI_BARRIER( MPI_COMM_WORLD, ierr )
if (myid==master) then;print*, "Calculating potential shift
s ...";endif
  potshaa=asisi*exp(-bsisi*rcoa)-csisi*rcoa**(-6) ! SiSi
interaction
  potshab=asina*exp(-bsina*rcoab)-csina*rcoab**(-6) ! SiNa
interaction
  potshac=asili*exp(-bsili*rcoac)-csili*rcoac**(-6) ! SiLi
interaction
  potshad=asio *exp(-bsio *rcoad)-csio *rcoad**(-6) ! SiO
interaction

  potshbb=anana*exp(-bnana*rcoab)-cnana*rcoab**(-6) ! NaNa
interaction
  potshbc=anali*exp(-bnali*rcoab)-cnali*rcoab**(-6) ! NaLi
interaction
  potshbd=anaoo *exp(-bnaoo*rcoab)-cnaoo *rcoab**(-6) ! NaO
interaction
  potshbe=ako *exp(-bko *rcoab)-cko *rcoab**(-6) ! Speci
al K-O interaction

  potshcc=alili*exp(-blili*rcoac)-clili*rcoac**(-6) ! LiLi
interaction
  potshcd=alio *exp(-blio*rcoac)-clio *rcoac**(-6) ! LiO
interaction

  potshdd=aooo *exp(-booo*rcoad)-coo *rcoad**(-6) ! OO
interaction
  call MPI_BARRIER( MPI_COMM_WORLD, ierr )

  if (myid==master) then;print*, "Calculating several constan
ts ...";endif
  twopi=2.d0*pi
  twopidlbox=twopi/lbox
  eps0=8.8542d0
  e2n=1602.19d0/(4.d0*pi*eps0)
  beta=alpha/lbox
  twoadrpi=2.d0*alpha/sqrt(pi)
  twobdpi=2.d0*beta/sqrt(pi)
  alpha2=alpha*alpha
  malpha=(-1.d0)*alpha2
  mbeta=(-1.d0)*beta*beta
  erkon=0.3275911d0*alpha
  erkonbeta=0.3275911d0*beta
  wskinwd2=(rcow*lbox*(rskinw-1.d0)/2.d0)**2 ! for the neigh
list

! Now we search for the maximum of all interactions that:
! - either contain a cij/r^6-Term
! - or interact with oxygen (negative charge).
! In both cases the potential will go to -infinity for r->
0, after climbing over a local maximum. Our goal is
! to find this maximum, and at about its peak fit an repul
sive d*r^-12 Term (d=1). This is to ensure that no particle
! gets caught at r=0.
  if (myid==master) then; print*, "Calculating potential exte
nsions (starting at 0.01A)..."; endif
  call repulsive_fit('Si-Si', chsi, chsi, asisi, bsisi, csisi, pot
shaa, rcoa, rcpolyaa, rcpolyaasq, rcpolyshaa)
  call repulsive_fit('Si-K ', chsi, chna, asina, bsina, csina, pot
shab, rcoab, rcpolyab, rcpolyabsq, rcpolyshab)
  call repulsive_fit('Si-Li', chsi, chli, asili, bsili, csili, pot
shac, rcoac, rcpolyac, rcpolyacsq, rcpolyshac)
  call repulsive_fit('Si-O ', chsi, cho, asio, bsio, csio, pot
shad, rcoad, rcpolyad, rcpolyadsq, rcpolyshad)
  call repulsive_fit('K -K ', chna, chna, anana, bnana, cnana, pot
shbb, rcobb, rcpolybb, rcpolybbsq, rcpolyshbb)

```

```

call repulsive_fit('K -Li', chna, chli, anali, bnali, cnali, pot
shbc, rcobc, rcpolybc, rcpolybcsq, rcpolyshbc)
if (myid==master) then; print*, "WARNING: K-O interaction fo
rcibly reset!!!!!!"; endif
call repulsive_fit('K -Os', chna, cho, ako, bko, cko, pot
shbe, rcobd, rcpolybe, rcpolybesq, rcpolyshbe)
call repulsive_fit('K -O ', chna, cho, anao, bnao, cnao, pot
shbd, rcobd, rcpolybd, rcpolybdsq, rcpolyshbd)
call repulsive_fit('Li-Li', chli, chli, alili, blili, clili, pot
shcc, rcocc, rcpolycc, rcpolyccsq, rcpolyshcc)
call repulsive_fit('Li-O ', chli, cho, alio, blio, clio, pot
shcd, rcocd, rcpolycd, rcpolycdsq, rcpolyshcd)
call repulsive_fit('O -O ', cho, cho, aoo, boo, coo, pot
shdd, rcodd, rcpolydd, rcpolyddsq, rcpolyshdd)

! Compute the times when the correlation functions are eva
luated
!-----
! Compute the times of the small correlation function

p=ncorrel
p=p-1
do k=1,p
  cortim(k)=nint(dfloat(ndumpstep)**(k/dfloat(p+1)))
enddo

cortim(1)=0
k=2
j=2

do while (j <= p)
  if(cortim(j) /= cortim(k)) then
    k=k+1
    cortim(k)=cortim(j)
    j=j+1
  else
    j=j+1
  endif
enddo

ncorrelnew=k

if(myid .eq. master) then
  write(071,*) 'ncorrelnew=', ncorrelnew
  do k=1, ncorrelnew
    write(071,*) cortim(k)
  enddo
endif

call MPI_BARRIER( MPI_COMM_WORLD, ierr )

if(ncorrelnew+ncorrelmax+1 > corldim+cor2dim) then
  if(myid.eq.master) then
    print*, 'error5', ncorrel, ncorrelnew, ncorrelmax, corldi
m, cor2dim
    write(071,*) 'error5', ncorrel, ncorrelnew, ncorrelmax
, corldim, cor2dim
  endif
  stop
endif

call MPI_BARRIER( MPI_COMM_WORLD, ierr )

! Random number generator:
!-----
!
! Generate the 24 seeds needed by the rng using a bad gene
rator but
! which is good enough for this purpose. See f. james Comp
. phys.
! comm. 60, 329, (1990). s(n+1)=mod(23*s(n), 10**8+1)
!
! To make sure that different system sizes imply different
initial
! configurations, the seeds depend on npart.

if(restart == 0) then

  iseed=olseed*myid
  do i=24, 1, -1
    iseed=abs(iseed-npart*182+iseed/npart)
    iseed=mod(iseed, 100000001)
    iseedn=iseed
    do j=1, 22 ! to prevent overflow we add 'by hand'
      iseedn=iseedn+iseed
      if(iseedn > 100000001) iseedn=iseedn-100000001
    enddo
    iseed=iseedn
    seed(i)=abs(mod(iseed, twop24))
  enddo

enddo

! Run the rng for a while to get rid of the initial val
ues

do j=1, 10000
  uni=seed(i24)-seed(j24)-carry
  if(uni < 0) then
    uni=uni+twop24
    carry=1
  else
    carry=0
  endif
  seed(i24)=uni
  i24=i24-1
  if(i24 == 0) i24=24
  j24=j24-1
  if(j24 == 0) j24=24
enddo

endif ! restart=0

! Start configuration:
!-----
!
! If this is a new run (olddat=0) generate a start configu
ration
! (positions and velocities; in case of an "old" run
! a start configuration in ascii format (restart=0) or in
! binary format with the start parameters (restart=1)
! has to be read in.

if(olddat == 0) then ! this is a new run

  ! Generate a random initial configuration
  do i=1, npart
    do j=1, 3
      uni=seed(i24)-seed(j24)-carry
      if(uni < 0) then
        uni=uni+twop24
        carry=1
      else
        carry=0
      endif
      seed(i24)=uni
      i24=i24-1
      if(i24 == 0) i24=24
      j24=j24-1
      if(j24 == 0) j24=24
      randn(j)=uni
    enddo
    pos(3*i-2)=randn(1)*lbox*intwop24
    pos(3*i-1)=randn(2)*lbox*intwop24
    pos(3*i)=randn(3)*lbox*intwop24
  enddo

  do i=1, 3*nlocal
    pindex=i+3*myid*nlocal
    posloc(i)=pos(pindex)
  enddo

  call maxbol(velloc, temp, m)
  volvel=0.d0

else ! new or old run?

  if(myid.eq.master) then
    write(6, fmt='('' File with the old data ? '', $)')
    read(060, fmt='(a)') filein2
  endif
  call MPI_BARRIER( MPI_COMM_WORLD, ierr )

! This is an old run - read in the old configuration

if(restart == 0) then

  if(myid.eq.master) then
    open(062, file=filein2, status='old', form='formatted
')

! Read only the old positions and velocities

    read(062, *) lbox, volvel
    do i=1, npart
      read(062, *) pos(3*i-2), pos(3*i-1), pos(3*i), vel(
3*i-2), vel(3*i-1), vel(3*i)
    enddo
    close(062)
  endif ! master

  call MPI_BARRIER( MPI_COMM_WORLD, ierr )
  call MPI_BCAST(lbox, 1, MPI_DOUBLE, master, MPI_COMM

```

```

_WORLD, ierr)
call MPI_BCAST(volvel, 1, MPI_DOUBLE, master, MPI_CO
MM_WORLD, ierr)
call MPI_BCAST(pos, 3*npart, MPI_DOUBLE, master, MPI
_COMM_WORLD, ierr)
call MPI_SCATTER(vel, 3*nlocal, MPI_DOUBLE, velloc,
3*nlocal, MPI_DOUBLE, master, MPI_COMM_WORLD, ierr)

do i=1,3*nlocal
  pindex=i+3*myid*nlocal
  posloc(i)=pos(pindex)
  runposloc(i)=posloc(i)
enddo

else ! then restart=1

  if(myid == master) then
    open(062,file=filein2,status='unknown',form='forma
tted')

    ! Read all the old information

    read(062,*) oolddat
    read(062,*) onapart; if(onapart /= nupart) then; p
rint*,'onapart /= nupart',onapart,nupart; stop; endif
    read(062,*) onbpart; if(onbpart /= nbpart) then; p
rint*,'onbpart /= nbpart',onbpart,nbpart; stop; endif
    read(062,*) oncpart; if(oncpart /= ncpart) then; p
rint*,'oncpart /= ncpart',oncpart,ncpart; stop; endif
    read(062,*) ondpart; if(ondpart /= ndpart) then; p
rint*,'ondpart /= ndpart',ondpart,ndpart; stop; endif
    read(062,*) orcoaa; if(orcoaa /= rcoaa) then; p
rint*,'orcoaa /= rcoaa',orcoaa,rcoaa ; stop; endif
    read(062,*) orcoab; if(orcoab /= rcoab) then; p
rint*,'orcoab /= rcoab',orcoab,rcoab ; stop; endif
    read(062,*) orcoac; if(orcoac /= rcoac) then; p
rint*,'orcoac /= rcoac',orcoac,rcoac ; stop; endif
    read(062,*) orcob; if(orcob /= rcob) then; p
rint*,'orcob /= rcob',orcob,rcob ; stop; endif
    read(062,*) orcobc; if(orcobc /= rcobc) then; p
rint*,'orcobc /= rcobc',orcobc,rcobc ; stop; endif
    read(062,*) orcoc; if(orcoc /= rcoc) then; p
rint*,'orcoc /= rcoc',orcoc,rcoc ; stop; endif
    read(062,*) orcod; if(orcod /= rcod) then; p
rint*,'orcod /= rcod',orcod,rcod ; stop; endif
    read(062,*) orcodd; if(orcodd /= rcodd) then; p
rint*,'orcodd /= rcodd',orcodd,rcodd ; stop; endif
    read(062,*) orcow; if(orcow /= rcow) then; p
rint*,'orcow /= rcow',orcow,rcow ; stop; endif
    read(062,*) orskinw; if(orskinw /= rskinw) then; p
rint*,'orskinw /= rskinw',orskinw,rskinw; stop; endif
    read(062,*) temp; temp=temp/11603.78d0
    read(062,*) ocooltim
    read(062,*) opresext
    if(opresext /= presext) then; print*,'opresext
/= presext',opresext,presext; stop; endif
    read(062,*) ompiston; if(ompiston /= mpiston) then
; print*,'ompiston /= mpiston',ompiston,mpiston; stop; endif

    read(062,*) otbathstep
    read(062,*) ocoolstep
    read(062,*) ocoolrate; ocoolrate=ocoolrate/11603.7
8d0
    if(ocoolrate /= coolrate) then; print*,'ocoolrate
/= coolrate',ocoolrate,coolrate; stop; endif
    read(062,*) onomtoten
    read(062,*) oolseed
    read(062,*) ohstep; if(ohstep /= hstep) then; prin
t*,'ohstep /= hstep',ohstep,hstep; stop; endif
    read(062,*) ontotstep
    read(062,*) onprintstep
    if(onprintstep /= nprintstep) then; print*,'onprin
tstep /= nprintstep',onprintstep,nprintstep; stop; endif
    read(062,*) ondumpstep
    if(ondumpstep /= ndumpstep) then; print*,'ondumpst
ep /= ndumpstep',ondumpstep,ndumpstep; stop; endif
    read(062,*) oncorrelmax
    read(062,*) oncorrel
    read(062,*) onsystdump
    read(062,fmt='(a)') string1
    read(062,fmt='(a)') fileout3
    read(062,*) starttime
    read(062,*) ncorrelnew
    read(062,*) decay
    read(062,*) decayew
    read(062,*) nvelstep

    ! Read positions and velocities

    read(062,*) lbox,volvel,vol

```

```

do i=1,npart
  read(062,*) pos(3*i-2),pos(3*i-1),pos(3*i),&
& vel(3*i-2),vel(3*i-1),vel(3*i), &
& runpos(3*i-2),runpos(3*i-1),runpos(
3*i)
enddo

! Random number variables

read(062,*) uni,carry,i24,j24
do i=1,24
  read(062,*) seed(i)
enddo
close(062)

endif ! master

!
! Send the data we have read from file 062 from process 0
! (master) to the other processes.
!
call MPI_BARRIER( MPI_COMM_WORLD, ierr )
call MPI_BCAST(oolddat, 1, MPI_INTEGER, master, MP
I_COMM_WORLD, ierr)
call MPI_BCAST(onapart, 1, MPI_INTEGER, master, MP
I_COMM_WORLD, ierr)
call MPI_BCAST(onbpart, 1, MPI_INTEGER, master, MP
I_COMM_WORLD, ierr)
call MPI_BCAST(oncpart, 1, MPI_INTEGER, master, MP
I_COMM_WORLD, ierr)
call MPI_BCAST(ondpart, 1, MPI_INTEGER, master, MP
I_COMM_WORLD, ierr)
call MPI_BCAST(orcoaa, 1, MPI_DOUBLE, master, MPI_
COMM_WORLD, ierr)
call MPI_BCAST(orcoab, 1, MPI_DOUBLE, master, MPI_
COMM_WORLD, ierr)
call MPI_BCAST(orcoac, 1, MPI_DOUBLE, master, MPI_
COMM_WORLD, ierr)
call MPI_BCAST(orcob, 1, MPI_DOUBLE, master, MPI_
COMM_WORLD, ierr)
call MPI_BCAST(orcobc, 1, MPI_DOUBLE, master, MPI_
COMM_WORLD, ierr)
call MPI_BCAST(orcoc, 1, MPI_DOUBLE, master, MPI_
COMM_WORLD, ierr)
call MPI_BCAST(orcod, 1, MPI_DOUBLE, master, MPI_
COMM_WORLD, ierr)
call MPI_BCAST(orcodd, 1, MPI_DOUBLE, master, MPI_
COMM_WORLD, ierr)
call MPI_BCAST(orcow, 1, MPI_DOUBLE, master, MPI_C
OMM_WORLD, ierr)
call MPI_BCAST(orskinw, 1, MPI_DOUBLE, master, MPI
_COMM_WORLD, ierr)
call MPI_BCAST(temp, 1, MPI_DOUBLE, master, MPI_CO
MM_WORLD, ierr)
call MPI_BCAST(ocooltim, 1, MPI_INTEGER, master, M
PI_COMM_WORLD, ierr)
call MPI_BCAST(opresext, 1, MPI_DOUBLE, master, M
PI_COMM_WORLD, ierr)
call MPI_BCAST(ompiston, 1, MPI_DOUBLE, master, MP
I_COMM_WORLD, ierr)
call MPI_BCAST(otbathstep, 1, MPI_INTEGER, master,
MPI_COMM_WORLD, ierr)
call MPI_BCAST(ocoolstep, 1, MPI_INTEGER, master,
MPI_COMM_WORLD, ierr)
call MPI_BCAST(ocoolrate, 1, MPI_DOUBLE, master, M
PI_COMM_WORLD, ierr)
call MPI_BCAST(onomtoten, 1, MPI_DOUBLE, master, M
PI_COMM_WORLD, ierr)
call MPI_BCAST(oolseed, 1, MPI_INTEGER, master, MP
I_COMM_WORLD, ierr)
call MPI_BCAST(ohstep, 1, MPI_DOUBLE, master, MPI_
COMM_WORLD, ierr)
call MPI_BCAST(ontotstep, 1, MPI_INTEGER, master,
MPI_COMM_WORLD, ierr)
call MPI_BCAST(onprintstep, 1, MPI_INTEGER, master
, MPI_COMM_WORLD, ierr)
call MPI_BCAST(ondumpstep, 1, MPI_INTEGER, master,
MPI_COMM_WORLD, ierr)
call MPI_BCAST(oncorrelmax, 1, MPI_INTEGER, master
, MPI_COMM_WORLD, ierr)
call MPI_BCAST(onsystdump, 1, MPI_INTEGER, master,
MPI_COMM_WORLD, ierr)
call MPI_BCAST(starttime, 1, MPI_INTEGER, master,
MPI_COMM_WORLD, ierr)
call MPI_BCAST(ncorrelnew, 1, MPI_INTEGER, master,
MPI_COMM_WORLD, ierr)
call MPI_BCAST(decay, 1, MPI_DOUBLE, master, MPI_C
OMM_WORLD, ierr)

```

```

        call MPI_BCAST(decayew, 1, MPI_DOUBLE, master, MPI_
_COMM_WORLD, ierr)
        call MPI_BCAST(lbox, 1, MPI_DOUBLE, master, MPI_CO
MM_WORLD, ierr)
        call MPI_BCAST(volvel, 1, MPI_DOUBLE, master, MPI_
COMM_WORLD, ierr)
        call MPI_BCAST(vol, 1, MPI_DOUBLE, master, MPI_COM
M_WORLD, ierr)
        call MPI_BCAST(pos, 3*npart, MPI_DOUBLE, master, M
PI_COMM_WORLD, ierr)
        call MPI_BCAST(uni, 1, MPI_INTEGER, master, MPI_CO
MM_WORLD, ierr)
        call MPI_BCAST(carry, 1, MPI_INTEGER, master, MPI_
COMM_WORLD, ierr)
        call MPI_BCAST(i24, 1, MPI_INTEGER, master, MPI_CO
MM_WORLD, ierr)
        call MPI_BCAST(j24, 1, MPI_INTEGER, master, MPI_CO
MM_WORLD, ierr)
        call MPI_BCAST(seed, 24, MPI_INTEGER, master, MPI_
COMM_WORLD, ierr)
        call MPI_SCATTER(vel, 3*nlocal, MPI_DOUBLE, velloc
, 3*nlocal, MPI_DOUBLE, master, MPI_COMM_WORLD, ierr)
        call MPI_SCATTER(runpos, 3*nlocal, MPI_DOUBLE, run
posloc, 3*nlocal, MPI_DOUBLE, master, MPI_COMM_WORLD, ierr)

        do i=1,3*nlocal
            pindex=i+3*myid*nlocal
            posloc(i)=pos(pindex)
        enddo

        endif ! restart

        if(myid == master) then
            close(060)
        endif

    endif ! olddat

!*****
!
!           Start of MD-Algorithm
!
!*****
!*****

    if(myid.eq.master) then

        do i=1,3*npart
            if(pos(i) < 0.or.pos(i) > lbox) then
                print*,'Positions in starting configuration bigge
r than box',i,pos(i)
                write(071,*) 'Positions in starting configuration
bigger than box',i,pos(i)
                stop
            endif
        enddo

        if(restart == 0) then
            if(olddat == 0) filein2=' '
            fileout3=fileout1
            write(071,fmt='(a)') filein2
        endif

    endif ! master

    call MPI_BARRIER( MPI_COMM_WORLD, ierr )

!
!           First MD-step
!-----
!-----

! initial box length dependent variables

    if (restart.eq.0) then
        vol=lbox*lbox*lbox
    endif

    invol=1.d0/vol
    density=npart*invol
    lboxd2=lbox/2.d0
    inlbox=1.d0/lbox
    sqrcwald=a12d2*lbox*lbox
    rcwald=rcow*lbox
    twopidlbox=twopi/lbox
    wskinwd2=(rcow*lbox*(rskinw-1.d0)/2.d0)**2

    beta=alpha/lbox
    twobdpi=2.d0*beta/sqrt(pi)
    mbeta=(-1.d0)*beta*beta
    erkonbeta=0.3275911d0*beta

```

```

    call prepk
    call neilist(pos,runposloc,neighposloc)
    call force(posloc,pos,acc)
    call analysel(posloc,pos,velloc)

    insttemp=ekin*2.d0/3.d0
    density=npart*invol

    if(restart == 0) starttime=0

    if(myid == master) then

        if(restart == 0) then
            press=insttemp*density+virial*invol
            write(071,fmt='(g12.6,10g23.15)') &
                & starttime*hstep,insttemp,ekin+epot,epot,p
ress,&
                & pos(3),pos(6),vol,volvel,&
                & ekin+epot+(0.5*mpiston*volvel**2+pressext
*vol)/npart,&
                & temp*11603.78d0

            open(061,file=filein1,status='unknown')
            restart=1
            write(061,*) olddat,' olddat'
            write(061,*) npart,' npart=N_Si'
            write(061,*) nbpart,' nbpart=N_Alk1'
            write(061,*) ncpart,' ncpart=N_Alk2'
            write(061,*) ndpart,' ndpart=N_O'
            write(061,*) lbox,' lbox'
            write(061,*) rcoaa,' rcutoff SiS1'
            write(061,*) rcoab,' rcutoff SiAlk1'
            write(061,*) rcoac,' rcutoff SiAlk2'
            write(061,*) rcoab,' rcutoff Alk1Alk1'
            write(061,*) rcoab,' rcutoff Alk1Alk2'
            write(061,*) rcoab,' rcutoff Alk2Alk2'
            write(061,*) rcoab,' rcutoff SiO'
            write(061,*) rcoab,' rcutoff Alk1O'
            write(061,*) rcoab,' rcutoff Alk2O'
            write(061,*) rcoab,' rcutoff OO'
            write(061,*) rcoab,' rcutoff Ewald '
            write(061,*) rskinw,' rskinw'
            write(061,*) temp*11603.78d0,' temp in K'
            write(061,*) cooltim,' cooltim'
            write(061,*) pressext,' pressext'
            write(061,*) mpiston,' mpiston'
            write(061,*) tbathstep,' tbathstep'
            write(061,*) coolstep,' coolstep'
            write(061,*) coolrate*11603.78d0,' coolrate in K'
            write(061,*) nomtoto,' nomtoto'
            write(061,*) olseed,' olseed'
            write(061,*) hstep,' hstep'
            write(061,*) ntotstep,' ntotstep'
            write(061,*) nprintstep,' nprintstep'
            write(061,*) ndumpstep,' ndumpstep'
            write(061,*) naverstep,' naverstep'
            write(061,*) ncorrelmax,' ncorrelmax (=K)'
            write(061,*) ncorrel,' ncorrel (=P)'
            write(061,*) nsystdump,' nsystdump'
            write(061,*) restart,' restart'
            write(061,*) alpha,' alpha(*lbox) of Ewald sum,'
            write(061,*) rmax,' rmax of Ewald sum,'
            write(061,*) kmax,' kmax of Ewald sum,'
            write(061,*) decay,' decay'
            write(061,*) decayew,' decayew'
            write(061,fmt='(a)') dump_file_1
            write(061,fmt='(a)') vel_file_1
            write(061,*) nvelstep,' nvelstep'
            close(061)
        endif ! restart

    endif ! master

    call MPI_BARRIER( MPI_COMM_WORLD, ierr )
    call MPI_GATHER( runposloc, 3*nlocal, MPI_DOUBLE, runpos,
3*nlocal, MPI_DOUBLE, master, MPI_COMM_WORLD, ierr )

! write the dump file at t=starttime

    if(myid == master) then

        dump_file_2=dump_file_1
        call addnumtostring(dump_file_2,starttime)

        open(075,file=dump_file_2,status='new',form='formatted'
)
        write(075,*) starttime
        do j=1,3*npart,3
            write(075,fmt='(3g25.15)') runpos(j),runpos(j+1),run
pos(j+2)
        enddo
        close(075)

        vel_file_2=vel_file_1

```

```

call addnumtostring(vel_file_2,starttime)

open(079,file=vel_file_2,status='new',form='formatted')

write(079,*) starttime
do j=1,3*npart,3
  write(079,fmt='(3g25.15)') vel(j),vel(j+1),vel(j+2)
enddo
close(079)

! Get time to ensure timely exit
begintime2 = MPI_WTIME()

endif ! master

! -----
! Time-loop
! -----
call MPI_BARRIER( MPI_COMM_WORLD, ierr )
if (myid==master) then
  print*,"#####"
  print*,"# Starting MD-Loop"
  print*,"#"
  print*,"#####"
endif
call MPI_BARRIER( MPI_COMM_WORLD, ierr )

do time=starttime+1,ntotstep
  !
  ! update the volume
  !
  pkin=0.d0
  pkinloc=0.d0
  do i=1,nlocal
    pkinloc=pkinloc+( velloc(3*i-2)*velloc(3*i-2)&
      & +velloc(3*i-1)*velloc(3*i-1)&
      & +velloc(3*i) *velloc(3*i) )*m(i)*
inlbox*inlbox
  enddo
  call MPI_ALLREDUCE( pkinloc, pkin, 1, MPI_DOUBLE, MPI_S
UM, MPI_COMM_WORLD, ierr )

  press=oned3*pkin*inlbox+virial*invol
  deltap=(press-pressex)*inmpiston
  vpart1=volvel*invol
  vol=vol + hstep*volvel + hsqd2*deltap
  !
  ! update volume dependent variables
  !
  olbox = lbox
  lbox = vol ** oned3
  lboxd2 = lbox * 0.5d0
  inlbox = 1.d0 / lbox
  invol = inlbox * inlbox * inlbox
  density= npart * invol
  twopidlbox=twopi/lbox
  sqrcwald=al2d2*lbox*lbox
  rcwald=rcow*lbox

  beta=alpha/lbox
  twobdpi=2.d0*beta/sqrt(pi)
  mbeta=(-1.d0)*beta*beta
  erkonbeta=0.3275911d0*beta
  wskinwd2=(rcow*lbox*(rskinw-1.d0)/2.d0)**2

  postmp = lbox/olbox
  do i=1,3*nlocal
    displa=hstep*velloc(i)+hsqd2*acc(i)-hsqd3*velloc(i)*
volvel*invol
    posdisp= displa*postmp
    posloc(i) = (posloc(i)+displa)* postmp
    runposloc(i)= runposloc(i) + posdisp
    fpart1(i) = acc(i)/olbox

    if(posloc(i) < 0.d0) then
      posloc(i)=posloc(i)+lbox
      if(posloc(i) < 0.d0) then
        print*,'ERROR in periodic boundary conditions!'
      ,
      ,
        print*,'Particle moved more than 1 box length.'
      ,
        print*,'Index: ',i
        print*,'Position: ',posloc(i)
      endif
    else
      if(posloc(i) > lbox) then
        posloc(i)=posloc(i)-lbox
        if(posloc(i) > lbox) then
          print*,'ERROR in periodic boundary conditio

```

```

ns!'
      print*,'Particle moved more than 1 box leng
th.'
      print*,'Index: ',i
      print*,'Position: ',posloc(i)
    endif
  endif
enddo
call MPI_ALLGATHER( posloc, 3*nlocal, MPI_DOUBLE, pos,
3*nlocal, MPI_DOUBLE, MPI_COMM_WORLD, ierr )

!
! Update of neighbor list if necessary, i.e. if any par
ticle moved
! further than half the skin-width:
!
i=1; newnl=0
do while (i <= nlocal .and. newnl <1)
  dx=runposloc(3*i-2)-neighposloc(3*i-2)
  dy=runposloc(3*i-1)-neighposloc(3*i-1)
  dz=runposloc(3*i) -neighposloc(3*i)
  rsq=dx*dx+dy*dy+dz*dz
  if (rsq >= wskinwd2) then
    newnl=1
  endif
  i=i+1
enddo

call MPI_ALLREDUCE( newnl, new_all, 1, MPI_INTEGER, MPI
_SUM, MPI_COMM_WORLD, ierr )

if (new_all > 0) then
  call neilist(pos,runposloc,neighposloc)
endif

!
! approximate volume velocities
!
call force(posloc,pos,acc)
press=oned3*pkin*inlbox+virial*invol
volvelapp = volvel + hstepd2*(deltap + (press-pressex)
*inmpiston )

!
! update position velocities
!
vpart2 = volvelapp * invol
do i =1,3*nlocal
  fpart2=acc(i)*inlbox
  velloc(i)=velloc(i)*postmp&
    & +( hstepd2*(fpart1(i)+fpart2)-hstepd3*vel
loc(i)*inlbox*(vpart1+vpart2))*(1.d0-hstepd3*vpart2) *lbox
enddo

!
! update the volume velocity
!
pkin=0.d0
pkinloc=0.d0
do i=1,nlocal
  pkinloc=pkinloc+( velloc(3*i-2)*velloc(3*i-2)&
    & +velloc(3*i-1)*velloc(3*i-1)&
    & +velloc(3*i) *velloc(3*i) )*m(i)*
inlbox*inlbox
enddo

call MPI_ALLREDUCE( pkinloc, pkin, 1, MPI_DOUBLE, MPI_S
UM, MPI_COMM_WORLD, ierr )

press=oned3*pkin*inlbox+virial*invol
volvel = volvel+hstepd2*(deltap +(press-pressex)*inmpi
ston)

!
! now md step is done. analyse some or influence the sy
stem if needed.
!
! Adjust the temperature ?
!
if(time <= cooltim .and. mod(time,coolstep) == 0) then
  temp=temp-deltat
  if(temp < 0.d0) then
    temp=0.d0
    coolstep=10000
    tbathstep=10
  endif
endif

!
! Thermalize the velocities?

```

```

!
if(time <= cooltim .and. mod(time,tbathstep) == 0) then
    call maxbol(velloc,temp,m)

    if(time+tbathstep > cooltim) then
        ! Last time of cooling. Adjust the total energy?
        if(nomtoten > -100) then
            call analyse1(posloc,pos,velloc)
            scafac=sqrt((nomtoten-epot)/ekin)
            do i=1,3*nlocal
                velloc(i)=velloc(i)*scafac
            enddo
        endif
    endif
endif

!
! Print the status of the system?
!
if(mod(time,nprintstep) == 0 .and. fileout1 /= ' ') the
n
    call analyse1(posloc,pos,velloc)
    if(myid == master) then
        insttemp=ekin*2.d0/3.d0
        press=insttemp*density+virial*invol
        write(071,fmt='(g12.6,10g23.15)') &
            & time*hstep*1.0217d0/100.d0, &
            & [ps]
        ! time
        & insttemp*11603.78d0, &
        ! temp (kin) [K]
        & ekin+epot, &
        ! E(total) [eV]
        & epot, &
        ! E(kin) [eV]
        & press*160.219d0, &
        ! pressure [GPa]
        & pos(3), &
        !
        & pos(6), &
        ! ?
        & vol, &
        ! vol [1/A^3]
        & volvel, &
        ! volvel
        & ekin+epot+ &
        & (0.5*mpiston*volvel**2+pressex*vol)/
npart,& ! ?
        & temp*11603.78d0
        ! set temp [K]
    endif ! master
    call MPI_BARRIER( MPI_COMM_WORLD, ierr )
endif

!
! Make a dump of the system in case of a crash?
!
if(mod(time,nsystdump) == 0 .and. fileout2 /= ' ') then

    call MPI_BARRIER( MPI_COMM_WORLD, ierr )
    call MPI_GATHER( velloc, 3*nlocal, MPI_DOUBLE, vel, 3
*nlocal, MPI_DOUBLE, master, MPI_COMM_WORLD, ierr )
    call MPI_GATHER( runposloc, 3*nlocal, MPI_DOUBLE, runpo
s, 3*nlocal, MPI_DOUBLE, master, MPI_COMM_WORLD, ierr )

    if(myid == master) then
        call write_dumpfile
    endif ! master

    call MPI_BARRIER( MPI_COMM_WORLD, ierr )

endif ! end of crash dump

call MPI_GATHER( velloc, 3*nlocal, MPI_DOUBLE, vel, 3*n
local, MPI_DOUBLE, master, MPI_COMM_WORLD, ierr )
call MPI_GATHER( runposloc, 3*nlocal, MPI_DOUBLE, runpo
s, 3*nlocal, MPI_DOUBLE, master, MPI_COMM_WORLD, ierr )

!
! the velocities
!
if(myid == master) then
    if(mod(time,nvelstep) == 0) then
        vel_file_2=vel_file_1
        call addnumtostring(vel_file_2,time)
        open(079,file=vel_file_2,status='new',form='forma
tted') !!!
        write(079,*) time
        do j=1,3*npart,3
            write(079,fmt='(6g25.15)') vel(j),vel(j+1),vel
(j+2),pos(j),pos(j+1),pos(j+2)
            enddo
            close(079)
        endif

        ! the positions
        do i=1,ncorrelnew

            if(mod(time,ndumpstep) == cortim(i).or.&
                & (mod(time,naverstep) ==&
                & mod(cortim(i),naverstep) .and
.&
                & time >= cortim(i)))
            then

                ! Write a configuration file:
                dump_file_2=dump_file_1
                call addnumtostring(dump_file_2,time)

                open(075,file=dump_file_2,status='new',form='f
ormatted') !!!
                write(075,*) time
                do j=1,3*npart,3
                    write(075,fmt='(3g25.15)') runpos(j),runpos
(j+1),runpos(j+2)
                enddo
                close(075)

                goto 8000

            endif
            enddo

8000 do i=1,3*npart
        if(pos(i) < 0.d0 .or. pos(i) > lbox) then
            print*,'error2',i,pos(i),vel(i),vol,volvel
        endif
    enddo

    endtime=MPI_WTIME()
    testtime=endtime-begintime

endif ! master

call MPI_BCAST( testtime, 1, MPI_DOUBLE_PRECISION, mas
ter, MPI_COMM_WORLD, ierr )

if(testtime > runlength) goto 9006

! -----
enddo ! time loop
! -----

! File with the final state

call MPI_BARRIER( MPI_COMM_WORLD, ierr )
call MPI_GATHER( velloc, 3*nlocal, MPI_DOUBLE, vel, 3*nlo
cal, MPI_DOUBLE, master, MPI_COMM_WORLD, ierr )

if(myid.eq.master) then
    endtime2 = MPI_WTIME()
    write(6,*) 'wct on process 0 = ',endtime2-begintime2

    if(fileout2 /= ' ') then
        open(072,file=fileout2,status='unknown',form='format
ted')
        write(072,*) lbox,volvel
        do i=1,npart
            write(072,fmt='(6g25.15)') pos(3*i-2),pos(3*i-1),
pos(3*i),vel(3*i-2),vel(3*i-1),vel(3*i)
        enddo
        close(072)
    endif

    print*,'good exit'
    close(071)

    endtime=MPI_WTIME()
    write(6,*) 'wall clock time of process 0 = ',endtime-be
gintime

endif ! master

call MPI_BARRIER( MPI_COMM_WORLD, ierr )
! print*,"Forcefit called ",forcefitcall," times."

```

```

call MPI_FINALIZE( ierr )

if (myid==0) then
  open(099,file='DONE',status='replace',form='formatted')
  close(099)
endif

stop 'final stop'

9006 if(myid.eq.master) close(071)

call MPI_GATHER( velloc, 3*nlocal, MPI_DOUBLE, vel, 3*nloc
al, MPI_DOUBLE, master, MPI_COMM_WORLD, ierr )
call MPI_GATHER( runposloc, 3*nlocal, MPI_DOUBLE, runpos,
3*nlocal, MPI_DOUBLE, master, MPI_COMM_WORLD, ierr )

if(myid.eq.master) then

  fileout1=fileout3
  call addnumtostring(fileout1,time)
  open(060,file=fileoffiles,status='unknown',form='formatt
ed')
  write(060,'(a)') filein1
  write(060,'(a)') fileout1
  write(060,'(a)') fileout2
  write(060,'(a)') fileout2
  close(060)
  write(6,*) 'time = ',time

  call write_dumpfile

endif ! master

if (myid==0) then
  open(098,file='DUMP',status='replace',form='formatted')
  close(098)
endif

call MPI_FINALIZE( ierr )

contains

subroutine write_dumpfile
  implicit none
  open(072,file=fileout2,status='unknown',form='formatted'
)
  write(072,*) olddat
  write(072,*) napart
  write(072,*) nbpart
  write(072,*) ncpart
  write(072,*) ndpart
  write(072,*) rcoaa
  write(072,*) rcoab
  write(072,*) rcoac
  write(072,*) rcoobb
  write(072,*) rcoobc
  write(072,*) rcoocc
  write(072,*) rcooad
  write(072,*) rcoabd
  write(072,*) rcoacd
  write(072,*) rcoodd
  write(072,*) rcow
  write(072,*) rskinv
  write(072,*) temp*11603.78d0
  write(072,*) cooltim
  write(072,*) pressext
  write(072,*) mpiston
  write(072,*) tbathstep
  write(072,*) coolstep
  write(072,*) coolrate*11603.78d0
  write(072,*) nomtoten
  write(072,*) olseed
  write(072,*) hstep
  write(072,*) ntotstep
  write(072,*) nprintstep
  write(072,*) ndumpstep
  write(072,*) ncorrelmax
  write(072,*) ncorrel
  write(072,*) nsystdump
  write(072,fmt='(a)') filein2
  write(072,fmt='(a)') fileout3
  write(072,*) time
  write(072,*) ncorrelnew
  write(072,*) decay
  write(072,*) decayew
  write(072,*) nvelstep
  ! Positions and velocities
  write(072,*) lbox,volvel,vol
  do i=1,npart
    write(072,fmt='(9g25.15)') pos(3*i-2) ,pos(3*i-1)
    ,pos(3*i),&

```

```

&
    vel(3*i-2) ,vel(3*i-1)
    ,vel(3*i),&
    &
    runpos(3*i-2),runpos(3*i-1)
  ) ,runpos(3*i)
  enddo
  ! Random number variables
  write(072,*) uni,carry,i24,j24
  do i=1,24
    write(072,*) seed(i)
  enddo
  close(072)
end subroutine write_dumpfile

subroutine repulsive_fit(name,chi,chj,A,B,C,potshift,rco,r
cpoly,rcpolysq,rcpolysh)
  implicit none
  real(DOUBLE) :: chi,chj,A,B,C,potshift,rco,rcpoly,rcpoly
sq,rcpolysq
  character(5) :: name
  start_rad=0.0001d0
  grain=0.0001d0
  if (myid==master) then
    print*,"#### ",name," #####"
    #####
  endif
  last_f_of_r=BKS_pot(chi,chj,start_rad+grain,A,B,C,potshi
ft,rco)
  new_f_of_r=BKS_pot(chi,chj,start_rad+grain*2.d0,A,B,C,po
tshift,rco)
  rad_count=2;
  if (new_f_of_r < last_f_of_r) then
    if (myid==master) then
      print*,"# Falling slope"
    endif
    rcpoly=0.d0
    rcpolysh=0.d0
  else
    do while (new_f_of_r >= last_f_of_r)
      last_f_of_r=new_f_of_r
      rad_count=rad_count+1
      if ((start_rad+grain*rad_count) >= rco) then
        print*,"# Maximum search failed!"
        stop
      endif
      new_f_of_r=BKS_pot(chi,chj,start_rad+grain*rad_cou
nt,A,B,C,potshift,rco)
    enddo
    rcpoly=start_rad+grain*rad_count
    rcpolysh=BKS_pot(chi,chj,rcpoly,A,B,C,potshift,rco)
    if (rcpolysh>1000.d0) then
      rcpoly=0.d0
      rcpolysh=0.d0
    endif
  endif
  rcpolysq=rcpoly*rcpoly
  if (myid==master) then
    if (rcpoly==0.d0) then
      print*,"# Maximum bigger than 1000.d0, poly-fit n
ot necessary."
    else
      print*,"# Maximum found at : ",rcpoly
      print*,"# Shift calculates to: ",rcpolysh
      print*,"# Grain count was : ",rad_count
    endif
    if (rcpoly==0.d0) then
      print*,"-> Formula for all r's : ",e2n,"*",chi,
      "*,chj,"/r+(",A,"*exp(-",&
      &
      B,"*r)-",C,""
      /((r**6)-",potshift,")*exp(-",&
      &
      decay,"/(r-",r
      ,rco,")^2)"
    else
      print*,"-> Formula for r<",rcpoly,":12.5*(r-",rcpo
ly,")^2+",rcpolysh
      print*,"-> Formula for r>",rcpoly,": ",e2n,"*",chi,
      "*,chj,"/r+(",A,"*exp(-",&
      &
      B,"*r)-",C,"/"
      r**6)-",potshift,")*exp(-",&
      &
      decay,"/(r-",r
      co,")^2)"
    endif
  endif
end subroutine repulsive_fit

function BKS_pot(chi,chj,r,A,B,C,potshift,rco) result(BKSp
ot) ! Small function returning Potential at r
  implicit none
  real(DOUBLE),intent(in) :: r,chi,chj,A,B,C,potshift,rco
  real(DOUBLE) :: BKSPot,potcont,potcont1
  potcont1=1.d0/(r-rco)
  potcont=exp(-decay*potcont1*potcont1)
  BKSPot=e2n*chi*chj/r+(A*exp(-B*r)-C/(r**6)-potshift)*pot
cont
end function BKS_pot

```





```

,ncpart))
select case (Select_Interaction(i,j,napart,nbpart
case(1) ! Si-Si interaction: ! heuer:1
call Precalc_Vars_1
call Precalc_Vars_ewald
tforce=chsisq*(twobdpi+tp*tmp-tp*dpotcontew)*t
mp*tmp*tp2*potcontew
tfcoul=tforce
vij=chsi*chsi*tp*exp(mbeta*rsq)*potcontew*tmp
if (rsq < rcoaasq) then
if (rij<=rcpolyaa) then
call Rsqfit_force(chsisq,rcpolyaa)
else
call BKS_force(rcoaa,asisi,bsisi,csisi,p
otshaa)
endif
endif
call Add_up_forces
case(2) ! Si-Na interaction: ! heuer:1
call Precalc_Vars_1
call Precalc_Vars_ewald
tforce=chsicchna*(twobdpi+tp*tmp-tp*dpotcontew)*
*tmp*tmp*tp2*potcontew
tfcoul=tforce
vij=chsi*chna*tp*exp(mbeta*rsq)*potcontew*tmp
if (rsq < rcoabsq) then
if (rij<=rcpolyab) then
call Rsqfit_force(chsicchna,rcpolyab)
else
call BKS_force(rcoab,asina,bsina,csina,p
otshab)
endif
endif
call Add_up_forces
case(3) ! Si-Li interaction: ! heuer:1
call Precalc_Vars_1
call Precalc_Vars_ewald
tforce=chsicchli*(twobdpi+tp*tmp-tp*dpotcontew)*
*tmp*tmp*tp2*potcontew
tfcoul=tforce
vij=chsi*chli*tp*exp(mbeta*rsq)*potcontew*tmp
if (rsq < rcoacsq) then
if (rij<=rcpolyac) then
call Rsqfit_force(chsicchli,rcpolyac)
else
call BKS_force(rcoac,asili,bsili,csili,p
otshac)
endif
endif
call Add_up_forces
case(4) ! Si-O interaction
call Precalc_Vars_1
call Precalc_Vars_ewald
tforce=chsicho*(twobdpi+tp*tmp-tp*dpotcontew)*
tmp*tmp*tp2*potcontew
tfcoul=tforce
vij=chsi*cho *tp*exp(mbeta*rsq)*potcontew*tmp
if (rsq < rcoadsq) then
if (rij<=rcpolyad) then
call Rsqfit_force(chsicho,rcpolyad)
else
call BKS_force(rcoad,asio,bsio,csio,pots
had)
endif
endif
call Add_up_forces
case(5) ! Na-Na interaction: ! heuer:1
call Precalc_Vars_1
call Precalc_Vars_ewald
tforce=chnasq*(twobdpi+tp*tmp-tp*dpotcontew)*t
mp*tmp*tp2*potcontew
tfcoul=tforce
vij=chna*chna*tp*exp(mbeta*rsq)*potcontew*tmp
if (rsq < rcobbsq) then
if (rij<=rcpolybb) then
call Rsqfit_force(chnasq,rcpolybb)
else
call BKS_force(rcobb,anana,bnana,cnana,p
otshbb)
endif
endif
call Add_up_forces
case(6) ! Na-Li interaction: ! heuer:1
call Precalc_Vars_1
call Precalc_Vars_ewald
tforce=chnachli*(twobdpi+tp*tmp-tp*dpotcontew)*
*tmp*tmp*tp2*potcontew
tfcoul=tforce
vij=chna*chli*tp*exp(mbeta*rsq)*potcontew*tmp
if (rsq < rcobcsq) then
if (rij<=rcpolybc) then
call Rsqfit_force(chnachli,rcpolybc)
else
call BKS_force(rcobc,anali,bnali,cnali,p
otshbc)
endif
endif
call Add_up_forces
case(7) ! Na-O interaction: ! heuer:1
call Precalc_Vars_1
call Precalc_Vars_ewald
tforce=chnacho*(twobdpi+tp*tmp-tp*dpotcontew)*
tmp*tmp*tp2*potcontew
tfcoul=tforce
vij=cho *chna*tp*exp(mbeta*rsq)*potcontew*tmp
if (rsq < rcobdsq) then
if (rij<=rcpolybe) then
call Rsqfit_force(chnacho,rcpolybe)
else
call BKS_force(rcobd,ako ,bko ,cko,potsh
be)
endif
endif
call Add_up_forces
case(8) ! Li-Li interaction: ! heuer:1
call Precalc_Vars_1
call Precalc_Vars_ewald
tforce=chlisq*(twobdpi+tp*tmp-tp*dpotcontew)*t
mp*tmp*tp2*potcontew
tfcoul=tforce
vij=chli*chli*tp*exp(mbeta*rsq)*potcontew*tmp
if (rsq < rcoccsq) then
if (rij<=rcpolycc) then
call Rsqfit_force(chlisq,rcpolycc)
else
call BKS_force(rcocc,alili,blili,clili,p
otshcc)
endif
endif
call Add_up_forces
case(9) ! Li-O interaction: ! heuer:1
call Precalc_Vars_1
call Precalc_Vars_ewald
tforce=chlichho*(twobdpi+tp*tmp-tp*dpotcontew)*
tmp*tmp*tp2*potcontew
tfcoul=tforce

```

```

        vij=cho *chli*tp*exp(mbeta*rsq)*potcontew*tmp
    if (rsq < rcocdsq) then
        if (rij<=rcpolycd) then
            call Rsqfit_force(chlich,rcpolycd)
        else
            call BKS_force(rcocd,alio,blio,clio,pots
hcd)
        endif
    endif
    call Add_up_forces
case(10) ! O=O interaction: ! heuer: ? polynom n
eeds refit
    call Precalc_Vars_1
    call Precalc_Vars_ewald
        tforce=chosq*(twobdpi+tp*tmp-tp*dpotcontew)*tm
p*tmp*tp2*potcontew
        tfcoul=tforce
        vij=cho *cho *tp*exp(mbeta*rsq)*potcontew*tmp
    if (rsq < rcocdsq) then
        if (rij<=rcpolydd) then
            call Rsqfit_force(chosq,rcpolydd)
        else
            call BKS_force(rcodd,aoo,boo,coo,potshdd
)
        endif
    endif
    call Add_up_forces
end select ! Select_Interaction(...)
vr=vr+vij
virmod=virmod+vij
endif ! rsq<srcewald
enddo ! loop over neighbors
fshortl(3*i-2)=fshortl(3*i-2)+fxi ! sum up force on pa
rticle i
fshortl(3*i-1)=fshortl(3*i-1)+fyi
fshortl(3*i )=fshortl(3*i )+fzi
enddo ! over particles
do i=1,numprocs
    recvcounts(i)=3*nlocal
enddo
call MPI_REDUCE_SCATTER( fshortl, fshort, recvcounts, MPI_
DOUBLE, MPI_SUM, MPI_COMM_WORLD, ierr )
call f_kwald( posloc,kvec,ckf,vcoulk )
tmp2=e2n*inlbox**2
do i=1,nlocal
    tmp4=3*i
    tmp3=1.d0/m(i)
    tmp=ckf(tmp4-2)*tmp2+fshort(tmp4-2)
    acc(tmp4-2)=tmp*tmp3
    tmp=ckf(tmp4-1)*tmp2+fshort(tmp4-1)
    acc(tmp4-1)=tmp*tmp3
    tmp=ckf(tmp4)*tmp2+fshort(tmp4)
    acc(tmp4)=tmp*tmp3
enddo
call MPI_ALLREDUCE( virshort, virshort_all, 1, MPI_DOUBLE,
MPI_SUM, MPI_COMM_WORLD, ierr )
call MPI_ALLREDUCE( virmod, virmod_all, 1, MPI_DOUBLE,MPI_
SUM, MPI_COMM_WORLD, ierr )
call MPI_ALLREDUCE( vr, vr_all, 1, MPI_DOUBLE,MPI_SUM, MPI_
_COMM_WORLD, ierr )
vcoul=(vr_all*lbox+vcoulk-vs)*inlbox*e2n
virial=(virshort_all+((vcoulk-vs)*inlbox+virmod_all)*e2n)/
3.d0
contains
subroutine Rsqfit_force(chij,rcpolyij)
    implicit none; real(DOUBLE) :: chij,rcpolyij
    forcefitcall=forcefitcall+1
    potl=rij-rcpolyij
    tfcoul=chij*(twobdpi+tp*tmp)*tmp*tmp*tp2
    tforce=tfcoul+(-25.d0*potl-chij*tmp*tmp)*tmp
    virshort=virshort -25.d0*potl*rij -chij*tmp
end subroutine Rsqfit_force
subroutine BKS_force(rcoi,j,A,B,C,potshij)
    implicit none; real(DOUBLE) :: rcoij,A,B,C,potshij,buf
    potcontl=1.d0/(rij-rcoi)
    potcont=exp(-decay*potcontl*potcontl)
    dpotcont=2.d0*decay*potcontl*potcontl*potcontl*potcont
oort4=tmp*tmp; oort4=oort4*oort4; oort4=oort4*oort4
    pot1=A*exp(-B*rij)
    pot2=C*oort4
    buf=(B*pot1*tmp-6.d0*pot2)*potcont -(potl-pot2*rsq-potsh
ij)*dpotcont*tmp
    tforce = tfcoul + buf
    virshort= virshort + buf*rsq
end subroutine BKS_force
subroutine Add_up_forces
    implicit none; integer(DEFAULT) :: idx; idx=3*j
    fxi=fxi+dx*tforce; fshortl(idx-2)=fshortl(idx-2)-dx*tfor
ce
    fyi=fyi+dy*tforce; fshortl(idx-1)=fshortl(idx-1)-dy*tfor
ce
    fzi=fzi+dz*tforce; fshortl(idx) =fshortl(idx) -dz*tfor
ce
end subroutine Add_up_forces
subroutine Precalc_Vars_1
    implicit none
    rij=sqrt(rsq)
    tp2=exp(mbeta*rsq)
    tmp=1.d0/rij
    t=1.d0/(1.d0+erkonbeta*rij)
    tp=t*(0.254829592d0+t*(-0.284496736d0+t*(1.421413741d0+t
*(-1.453152027d0+t*1.061405429d0 )))
end subroutine Precalc_Vars_1
subroutine Precalc_Vars_ewald
    implicit none
    potcontewl=1.d0/(rij-rcwald)
    potcontew=exp(-decayew*potcontewl*potcontewl)
    dpotcontew=2.d0*decayew*potcontewl*potcontewl*potcontewl
end subroutine Precalc_Vars_ewald
end subroutine force
subroutine analyse1(posloc,pos,velloc)
    use MYMPI
    use typeDefs
    use program_wide_constants
    use sim_commons
    use ewald,only: vs,kvec
    use charge_and_mass,only: chsi,cho,chna,ch,chli
    use potpar_commons
    ! use potential_extension
    implicit none
    real(DOUBLE) :: pos(partdim),posloc(partdimloc),velloc(par
tdimloc)
    integer(DEFAULT) :: i,j,pindex
    integer(DEFAULT) :: high,neighno
    real(DOUBLE) :: xx,yy,zz,rsq,rij,dx,dy,dz
    real(DOUBLE) :: ekinloc,epotloc
    real(DOUBLE) :: potcont ,potcontl ,dpotcont
    real(DOUBLE) :: virshort,virshort_all,potl,pot2
    real(DOUBLE) :: chsisq,chsichna,chsichli,chsicho,chnasq,ch
nachli,chnacho,chlisq,chlich,chosq
    real(DOUBLE) :: vr,vr_all,vcoulk,vcoul
    real(DOUBLE) :: virmod,virmod_all
    real(DOUBLE) :: tmp0,tmp1,tmp2
    integer(DEFAULT) :: Select_Interaction,idx_tmp
    ! compute kinetic energy
    ekinloc=0.d0
    do i=1,nlocal
        idx_tmp=3*i
        ekinloc=ekinloc+m(i)*(velloc(idx_tmp-2)*velloc(idx_tmp-
2)&
            &
            +velloc(idx_tmp-1)*velloc(idx_tmp-
1)&
            &
            +velloc(idx_tmp)*velloc(idx_tmp) )
    enddo
    ekinloc=ekinloc/2.d0/npart
    call MPI_ALLREDUCE( ekinloc, ekin, 1, MPI_DOUBLE,MPI_SUM,
MPI_COMM_WORLD, ierr )

```

```

! now compute the potential energy

epotloc=0.d0
chsisq=chsi*chsi; chsichna=chsi*chna; chsichli=chsi*chli;
chsicho=chsi*cho
chnasq =chna*chna; chnachli=chna*chli;
chnacho=chna*cho
chlichq =chli*chli;
chlichcho=chli*cho
chosq =cho *cho

do i=nstart,nend
  pindex=i-myd*nlocal
  xx=pos(3*i-2)
  yy=pos(3*i-1)
  zz=pos(3*i)
  high=1+neighwald(1,pindex)

  do neighno=2,high
    j=neighwald(neighno,pindex)
    dx=xx-pos(3*j-2)
    dy=yy-pos(3*j-1)
    dz=zz-pos(3*j)

    if(dx < -lboxd2) then; dx=dx+lbox; else; if(dx > lbo
xd2) dx=dx-lbox; endif
    if(dy < -lboxd2) then; dy=dy+lbox; else; if(dy > lbo
xd2) dy=dy-lbox; endif
    if(dz < -lboxd2) then; dz=dz+lbox; else; if(dz > lbo
xd2) dz=dz-lbox; endif
    rsq=dx*dx+dy*dy+dz*dz

    ! Select the kind of interaction (Si-Si, Si-Na, Si-L
i, Si-O,
! Na-Na, Na-Li, Na-O, Li-Li, Li-O, or O-O interactio
n):
    select case (Select_Interaction(i,j,npart,nbpart,nc
part))

! Calculate the energy depending on the kind of i
nteraction:

case(1) ! Si-Si interaction

if(rsq < rcoaasq) then
  rij=sqrt(rsq)
  if(rij <= rcpolyaa) then
    call Rmsqfit(chsisq,rcpolyshaa,rcpolyaa)
  else
    call BKS_pot(rcoaa,asisi,bsisi,csisi,potsha
a)
  endif
endif

case(2) ! Si-Na interaction

if(rsq < rcoabsq) then
  rij=sqrt(rsq)
  if(rij <= rcpolyab) then
    call Rmsqfit(chsichna,rcpolyshab,rcpolyab)
  else
    call BKS_pot(rcoab,asina,bsina,csina,potsha
b)
  endif
endif

case(3) ! Si-Li interaction

if(rsq < rcoacsq) then
  rij=sqrt(rsq)
  if(rij <= rcpolyac) then
    call Rmsqfit(chsichli,rcpolyshac,rcpolyac)
  else
    call BKS_pot(rcoac,asili,bsili,csili,potsha
c)
  endif
endif

case(4) ! Si-O interaction

if(rsq < rcoadsq) then
  rij=sqrt(rsq)
  if(rij <= rcpolyad) then
    call Rmsqfit(chsicho,rcpolyshad,rcpolyad)
  else
    call BKS_pot(rcoad,asio,bsio,csio,potshad)
  endif
endif

case(5) ! Na-Na interaction

if(rsq < rcobbsq) then
  rij=sqrt(rsq)
  if(rij <= rcpolybb) then
    call Rmsqfit(chnasq,rcpolyshbb,rcpolybb)
  else
    call BKS_pot(rcobb,anana,bnana,cnana,potshb
b)
  endif
endif

case(6) ! Na-Li interaction

if(rsq < rcobcsq) then
  rij=sqrt(rsq)
  if(rij <= rcpolybc) then
    call Rmsqfit(chnachli,rcpolyshbc,rcpolybc)
  else
    call BKS_pot(rcobc,anali,bnali,cnali,potshb
c)
  endif
endif

case(7) ! Na-O interaction

if(rsq < rcobdsq) then
  rij=sqrt(rsq)
  if(rij <= rcpolybe) then
    call Rmsqfit(chnacho,rcpolyshbe,rcpolybe)
  else
    call BKS_pot(rcobd,ako ,bko ,cko,potshbe)
  endif
endif

case(8) ! Li-Li interaction

if(rsq < rcoccsq) then
  rij=sqrt(rsq)
  if(rij <= rcpolycc) then
    call Rmsqfit(chlisq,rcpolyshcc,rcpolycc)
  else
    call BKS_pot(rcocc,alili,blili,clili,potshc
c)
  endif
endif

case(9) ! Li-O interaction

if(rsq < rcocdsq) then
  rij=sqrt(rsq)
  if(rij <= rcpolycd) then
    call Rmsqfit(chlichq,rcpolyshcd,rcpolycd)
  else
    call BKS_pot(rcocd,alio,blio,clio,potshcd)
  endif
endif

case(10) ! O-O interaction

if(rsq < rcoddsq) then
  rij=sqrt(rsq)
  if(rij <= rcpolydd) then
    call Rmsqfit(chosq,rcpolyshdd,rcpolydd)
  else
    call BKS_pot(rcodd,aoo,boo,coo,potshdd)
  endif
endif

end select ! Select_Interaction(...)

enddo ! over neighbours
enddo ! over particles

call rwald(pos,virmod,vr)
call kwald(posloc,vcoulk,kvec)

call MPI_ALLREDUCE( epotloc, epot, 1, MPI_DOUBLE,MPI_SUM,
MPI_COMM_WORLD, ierr )
call MPI_ALLREDUCE( vr, vr_all, 1, MPI_DOUBLE,MPI_SUM, MPI
_COMM_WORLD, ierr )

vcoul=(vr_all*lbox+vcoulk-vs)*inlbox*e2n
epot=epot+vcoul
epot=epot/npart

contains

subroutine Rmsqfit(chij,rcpolyshij,rcpolyij)
  implicit none;real(DOUBLE) :: chij,rcpolyshij,rcpolyij
  pot1=rij-rcpolyij
  pot2=chij*e2n/rij ! explicit Clb to take away, because it
will be added up later (r/k-wald)
  epotloc =epotloc + 12.5d0*pot1*pot1+rcpolyshij -pot2

```

```

end subroutine Rmsqfit

subroutine BKS_pot(rcoi,j,A,B,C,potshij)
  implicit none;real(DOUBLE) :: rcoij,A,B,C,potshij
  potcont1=1.d0/(rij-rcoi)
  potcont=exp(-decay*potcont1*potcont1)
  pot1=A*exp(-B*rij)
  pot2=C/(rsq*rsq*rsq)
  epotloc=epotloc+(pot1-pot2-potshij)*potcont
end subroutine BKS_pot

end subroutine analyse1

! The next subroutine calculates the neighbor list
! neighwald(neighbor number,part. number i) = part. number j
! where i and j are counted in the following way:
!
! -----
! | i | j | | i | j |
! -----
! | 1 | 2,N/2+1 | | N/2+1 | N/2+2,N |
! | 2 | 3,N/2+2 | | N/2+2 | N/2+3,N;1,1 |
! | 3 | 4,N/2+3 | | N/2+3 | N/2+4,N;1,2 |
! | . | . | | . | . |
! | . | . | | . | . |
! | . | . | | . | . |
! | N/2-1 | N/2,N-1 | | N-1 | N,N;1,N/2-2 |
! | N/2 | N/2+1,N | | N | 1,N/2-1 |
! -----

subroutine neilist(pos,runposloc,neighposloc)

  use MYMPI
  use typeDefs
  use program_wide_constants
  use sim_commons,only: vol,lbox,lboxd2,npart,neighwald,al2s
  kin

  implicit none

  real(DOUBLE) :: pos(partdim)
  real(DOUBLE) :: runposloc(partdimloc),neighposloc(partdimloc)
  integer(DEFAULT) :: i,j,k,pindex
  integer(DEFAULT) :: listend,countw
  real(DOUBLE) :: xx,yy,zz,dx,dy,dz
  real(DOUBLE) :: rsq,alsksq

  alsksq=al2skin*lbox*lbox

  do i=nstart,nend
    pindex=i-myid*nlocal
    countw=1
    xx=pos(3*i-2)
    yy=pos(3*i-1)
    zz=pos(3*i)

    if(i <= ceiling(npart/2.0)) then
      listend=ceiling(npart/2.0)
    else
      listend=ceiling(npart/2.0)-1
    endif

    do k=1,listend
      j=mod(i+k-1,npart)+1
      dx=xx-pos(3*j-2)
      dy=yy-pos(3*j-1)
      dz=zz-pos(3*j)

      if(dx < -lboxd2) then; dx=dx+lbox; else; if(dx > lboxd2) dx=dx-lbox; endif
      if(dy < -lboxd2) then; dy=dy+lbox; else; if(dy > lboxd2) dy=dy-lbox; endif
      if(dz < -lboxd2) then; dz=dz+lbox; else; if(dz > lboxd2) dz=dz-lbox; endif

      rsq=dx*dx+dy*dy+dz*dz

      if(rsq < alsksq) then !found neighb. for ewald
        countw=countw+1
        neighwald(countw,pindex)=j
      endif

    enddo ! k=1,listend

    neighwald(1,pindex)=countw-1

    if(myid.eq.master) then
      if(countw > 800) then
        print*,'errorwald3', countw
        write(071,*) 'errorwald3 ',countw
      endif
    endif
  enddo

```

```

  endif
endif
enddo

do i=1,3*nlocal
  neighposloc(i)=runposloc(i)
enddo

end subroutine neilist

subroutine maxbol(velloc,temp,m)

  use MYMPI
  use typeDefs
  use program_wide_constants
  use sim_commons,only: vol,lbox,lboxd2,npart,tdtp24,carry,
  i24,j24,seed,twop24,inlbox,napart,nbpart,ncpart,ndpart

  implicit none

  real(DOUBLE) :: velloc(partdimloc),m(nchpartloc)

  integer(DEFAULT) :: i,uni,pindex,t
  real(DOUBLE) :: temp,v1,v2,r,s,ma,mb,tmp,tm
  real(DOUBLE) :: scafac,ekin,ekinloc
  real(DOUBLE) :: cmvx(4),cmvy(4),cmvz(4)
  real(DOUBLE) :: cmvxloc(4),cmvyloc(4),cmvzloc(4)
  real(DOUBLE) :: nparts(4)

  nparts(1)=napart
  nparts(2)=nbpart
  nparts(3)=ncpart
  nparts(4)=ndpart

  do i=1,3*nlocal,2
    pindex=i+3*myid*nlocal
    10 uni=seed(i24)-seed(j24)-carry
    if(uni < 0) then
      uni=uni+twop24
      carry=1
    else
      carry=0
    endif
    seed(i24)=uni
    i24=i24-1
    if(i24 == 0) i24=24
    j24=j24-1
    if(j24 == 0) j24=24
    v1=tdtp24*(uni+1)-1.0

    uni=seed(i24)-seed(j24)-carry
    if(uni < 0) then
      uni=uni+twop24
      carry=1
    else
      carry=0
    endif
    seed(i24)=uni
    i24=i24-1
    if(i24 == 0) i24=24
    j24=j24-1
    if(j24 == 0) j24=24
    v2=tdtp24*(uni+1)-1.0

    s=v1*v1+v2*v2
    if(s >= 1.0) goto 10
    r=sqrt(-2.0*log(s)/s)

    velloc(i)=v1*r
    if(i < 3*nlocal) then
      velloc(i+1)=v2*r
    endif
  enddo

  ! Scale velocities with respective mass factor

  do i=1,nlocal
    tmp=1.d0/sqrt(m(i))
    velloc(3*i-2)=tmp*velloc(3*i-2)
    velloc(3*i-1)=tmp*velloc(3*i-1)
    velloc(3*i)=tmp*velloc(3*i)
  enddo

  ! total momentum = 0 for each particle type !!!!

  do t=1,4
    cmvxloc(t)=0.d0
    cmvyloc(t)=0.d0
    cmvzloc(t)=0.d0
  enddo

```

```

do i=1,nlocal
  tm=m(i)
  pindex=i+myid*nlocal
  if(pindex <= napart) then
    t=1
  else
    if(pindex <= napart+nbpart) then
      t=2
    else
      if(pindex <= napart+nbpart+ncpart) then
        t=3
      else
        t=4
      endif
    endif
  endif
  cmvxloc(t)=cmvxloc(t)+velloc(3*i-2)*tm
  cmvyloc(t)=cmvyloc(t)+velloc(3*i-1)*tm
  cmvzloc(t)=cmvzloc(t)+velloc(3*i) *tm
enddo

do t=1,4
  if (nparts(t).ne.0) then
    call MPI_BARRIER( MPI_COMM_WORLD, ierr )
    call MPI_ALLREDUCE( cmvxloc(t), cmvx(t), 1, MPI_DOU
LE, MPI_SUM, MPI_COMM_WORLD, ierr )
    call MPI_ALLREDUCE( cmvyloc(t), cmvy(t), 1, MPI_DOU
LE, MPI_SUM, MPI_COMM_WORLD, ierr )
    call MPI_ALLREDUCE( cmvzloc(t), cmvz(t), 1, MPI_DOU
LE, MPI_SUM, MPI_COMM_WORLD, ierr )
    cmvx(t)=cmvx(t)/nparts(t)
    cmvy(t)=cmvy(t)/nparts(t)
    cmvz(t)=cmvz(t)/nparts(t)
  endif
enddo

do i=1,nlocal
  tm=1.d0/m(i)
  pindex=i+myid*nlocal
  if(pindex <= napart) then
    t=1
  else
    if(pindex <= napart+nbpart) then
      t=2
    else
      if(pindex <= napart+nbpart+ncpart) then
        t=3
      else
        t=4
      endif
    endif
  endif
  velloc(3*i-2)=velloc(3*i-2)-cmvx(t)*tm
  velloc(3*i-1)=velloc(3*i-1)-cmvy(t)*tm
  velloc(3*i) =velloc(3*i) -cmvz(t)*tm
enddo

! Scale velocities to get the right temperature

ekinloc = 0.d0
do i=1,nlocal
  ekinloc=ekinloc+m(i)*(velloc(3*i-2)*velloc(3*i-2)&
& +velloc(3*i-1)*velloc(3*i-1)&
& +velloc(3*i) *velloc(3*i) )
enddo

call MPI_BARRIER( MPI_COMM_WORLD, ierr )
call MPI_ALLREDUCE( ekinloc, ekin, 1, MPI_DOUBLE, MPI_SUM,
MPI_COMM_WORLD, ierr )

scafac=sqrt(3.d0*temp*npart/ekin)

do i=1,3*nlocal
  velloc(i)=velloc(i)*scafac
enddo

return
end subroutine maxbol

! Routine to set up the wave-vectors for the Ewald sum.
! see Allen & Tildesley: f22.for
! k for box of unit length
! & selfterm vs included , because also once at the beginni
ng calc.
! in : ch,alpha out: kvec,vs

subroutine prepk

  use MYMPI
  use typeDefs
  use program_wide_constants
  use sim_commons,only: vol,lbox,lboxd2,npart,napart,nbpart,
ncpart,ndpart,neighwald
  use charge_and_mass,only: chsi,cho,chna,chli,ch
  use rsum_commons,only: rmax,rsqmax,mbeta,twobdpi,erkonbeta
,sqrcewald,rcewald,decayew

  implicit none

  real(DOUBLE) :: pos(partdim)
  integer(DEFAULT) :: i,j,pindex
  integer(DEFAULT) :: high,neighno
  real(DOUBLE) :: invrij,vijtmp,vij,vr
  real(DOUBLE) :: rxi,ryi,rzi,rxi,ryij,rzij,rijsq,rij
  real(DOUBLE) :: t,tp
  real(DOUBLE) :: virmod,virij
  real(DOUBLE) :: potcontew1,potcontew,dpotcontew

  use ewald
  use charge_and_mass,only: chsi,cho,chna,ch
  use rsum_commons,only: twopi,twopidlbox
  use ksum_commons,only: kmax,ksqmax,kx_field,ky_field,kz_fi
eld

  implicit none

  integer(DEFAULT) :: i,k
  integer(DEFAULT) :: ksq,kx,ky,kz,kcount!,totk
  real(DOUBLE) :: b,rkx,rky,rkz,rksq
  real(DOUBLE) :: vs_loc

  b=1.d0/4.d0/alpha/alpha

  ! loop over k-vectors, note that k_x is pos. (sym.)

  totk=0
  kcount=0

  do kx=0,kmax
    rkx=twopi*dfloat(kx)
    do ky=-kmax,kmax
      rky=twopi*dfloat(ky)
      do kz=-kmax,kmax
        rkz=twopi*dfloat(kz)
        ksq=kx*kx+ky*ky+kz*kz
        kcount=kcount+1
        if((ksq < ksqmax) .and. (ksq /= 0)) then
          totk=totk+1
          if(totk > maxk) stop 'kvec is too small'
          rksq=rkx*rkx+rky*rky+rkz*rkz
          kvec(totk)=twopi*exp(-b*rksq)/rksq
          kx_field(totk)=kx
          ky_field(totk)=ky
          kz_field(totk)=kz
        endif
      enddo ! kz
    enddo ! ky
  enddo ! kx

  if(myid.eq.master) then
    write(*,' ( " EWALD sum SETUP COMPLETE " ) ')
    write(*,' ( " NUMBER OF WAVEVECTORS IS " , I5 ) ')
    totk
    write(*,' ( " NUMBER OF WAVEVECTORS SET TO " , I5 ) ')
    totk
  endif

  call MPI_BARRIER( MPI_COMM_WORLD, ierr )

  ! Calculate self term of k-sum.

  vs_loc=0.d0
  do i=1,nlocal
    vs_loc=vs_loc+ch(i)*ch(i)
  enddo

  call MPI_ALLREDUCE( vs_loc, vs, 1, MPI_DOUBLE, MPI_SUM, MP
I_COMM_WORLD, ierr )

  vs=alpha*vs/sqrt(pi)
end subroutine prepk

! real(DOUBLE) :: space part of potential energy by ewald me
thod
! in: ch,pos out: virmod,vr

subroutine rwald(pos,virmod,vr)

  use MYMPI
  use typeDefs
  use program_wide_constants
  use sim_commons,only: vol,lbox,lboxd2,npart,napart,nbpart,
ncpart,ndpart,neighwald
  use charge_and_mass,only: chsi,cho,chna,chli,ch
  use rsum_commons,only: rmax,rsqmax,mbeta,twobdpi,erkonbeta
,sqrcewald,rcewald,decayew

  implicit none

  real(DOUBLE) :: pos(partdim)
  integer(DEFAULT) :: i,j,pindex
  integer(DEFAULT) :: high,neighno
  real(DOUBLE) :: invrij,vijtmp,vij,vr
  real(DOUBLE) :: rxi,ryi,rzi,rxi,ryij,rzij,rijsq,rij
  real(DOUBLE) :: t,tp
  real(DOUBLE) :: virmod,virij
  real(DOUBLE) :: potcontew1,potcontew,dpotcontew

```

```

integer(DEFAULT) :: Select_Interaction

vr=0.d0
virmod=0.d0

do i=nstart,nend
  pindex=i-myid*nlocal
  rxi=pos(3*i-2)
  ryi=pos(3*i-1)
  rzi=pos(3*i)
  high=1+neighwald(1,pindex)

  do neighno=2,high
    j=neighwald(neighno,pindex)
    rxij=rxi-pos(3*j-2)
    ryij=ryi-pos(3*j-1)
    rzij=rzi-pos(3*j)

    if(rxij < -lboxd2) then; rxij=rxij+lbox; else; if(rx
ij > lboxd2) rxij=rxij-lbox; endif
    if(ryij < -lboxd2) then; ryij=ryij+lbox; else; if(ry
ij > lboxd2) ryij=ryij-lbox; endif
    if(rzij < -lboxd2) then; rzij=rzij+lbox; else; if(rz
ij > lboxd2) rzij=rzij-lbox; endif
    rijsq=rxij*rxij+ryij*ryij+rzij*rzij

    if(rijsq < sqrcewald) then

      rij=sqrt(rijsq)
      invrij=1.d0/rij
      t=1.d0/(1.d0+erkonbeta*rij)
      tp=t*(0.254829592d0+t*(-0.284496736d0+t*(1.421413
741d0+t*(-1.453152027d0+t*1.061405429d0 )))

      potcontew1=1.d0/(rij-rcewald)
      potcontew=exp(-decayew*potcontew1*potcontew1)

      ! Calculate the energie depending on the kind of
interaction:
      select case (Select_Interaction(i,j,napart,nbpart
,ncpart))

        case(1) ! Si-Si interaction

          vij=chsi*chsi*tp*exp(mbeta*rijsq)*potcontew*in
vrij
          virij=vij

        case(2) ! Si-Na interaction

          vij=chsi*chna*tp*exp(mbeta*rijsq)*potcontew*in
vrij
          virij=vij

        case(3) ! Si-Li interaction

          vij=chsi*chli*tp*exp(mbeta*rijsq)*potcontew*in
vrij
          virij=vij

        case(4) ! Si-O interaction

          vij=chsi*cho*tp*exp(mbeta*rijsq)*potcontew*in
vrij
          virij=vij

        case(5) ! Na-Na interaction

          vij=chna*chna*tp*exp(mbeta*rijsq)*potcontew*in
vrij
          virij=vij

        case(6) ! Na-Li interaction

          vij=chna*chli*tp*exp(mbeta*rijsq)*potcontew*in
vrij
          virij=vij

        case(7) ! Na-O interaction

          vij=cho*chna*tp*exp(mbeta*rijsq)*potcontew*in
vrij
          virij=vij

        case(8) ! Li-Li interaction

          vij=chli*chli*tp*exp(mbeta*rijsq)*potcontew*in
vrij
          virij=vij

        case(9) ! Li-O interaction

          vij=cho*chli*tp*exp(mbeta*rijsq)*potcontew*in
vrij
          virij=vij

          case(10) ! O-O interaction

          vij=cho*cho*tp*exp(mbeta*rijsq)*potcontew*in
vrij
          virij=vij

        end select ! Select_Interaction(...)

      vr=vr+vij
      virmod=virmod+virij

    endif

  enddo ! neighno
enddo ! i

end subroutine rwald

! K-Space part of potential energy by ewald method
! see Allen & T.: but * no self term here (only at beginning
)
! in: ch,pos,kvec out: vcoulk

subroutine kwald(posloc,vcoulk,kvec)

  use MYMPI
  use typeDefs
  use program_wide_constants
  use charge_and_mass,only: chsi,cho,chna,ch
  use rsum_commons,only: twopi,twopidlbox
  use ksum_commons,only: kmax,ksqmax,kx_field,ky_field,kz_fi
eld
  use ewald,only: totk

  implicit none

  real(DOUBLE) :: posloc(partdimloc),kvec(maxk)
  complex(DOUBLE) :: eikx(1:nchpartloc,0:kmax_para)
  complex(DOUBLE) :: eiky(1:nchpartloc,-kmax_para:kmax_para)

  complex(DOUBLE) :: eikz(1:nchpartloc,-kmax_para:kmax_para)

  complex(DOUBLE) :: eikr(nchpartloc),sum(maxk),sum_all(maxk)

  integer(DEFAULT) :: i,j,kcount,kx,ky,kz
  real(DOUBLE) :: vcoulk,fact

  ! construct exp(ikr) for all ions and k-vectors
  ! calculate kx, ky, kz= 0, -1 and 1 explicitly

  do i=1,nlocal
    eikx(i,0)=(1.d0,0.d0)
    eiky(i,0)=(1.d0,0.d0)
    eikz(i,0)=(1.d0,0.d0)
    eikx(i,1)=cmplx(cos(twopidlbox*posloc(3*i-2)),&
& sin(twopidlbox*posloc(3*i-2)))
    eiky(i,1)=cmplx(cos(twopidlbox*posloc(3*i-1)),&
& sin(twopidlbox*posloc(3*i-1)))
    eikz(i,1)=cmplx(cos(twopidlbox*posloc(3*i)),&
& sin(twopidlbox*posloc(3*i)))
    eiky(i,-1)=conjg(eiky(i,1))
    eikz(i,-1)=conjg(eikz(i,1))
  enddo

  ! calculate remaining kx, ky and kz by recurrence

  do j=2,kmax
    do i=1,nlocal
      eikx(i,j)=eikx(i,j-1)*eikx(i,1)
      eiky(i,j)=eiky(i,j-1)*eiky(i,1)
      eiky(i,-j)=conjg(eiky(i,j))
      eikz(i,j)=eikz(i,j-1)*eikz(i,1)
      eikz(i,-j)=conjg(eikz(i,j))
    enddo
  enddo

  ! sum over all vectors

  vcoulk=0.d0
  do kcount=1,totk !maxk
    kx=kx_field(kcount)
    ky=ky_field(kcount)
    kz=kz_field(kcount)
    sum(kcount)=(0.d0,0.d0)
    do i=1,nlocal
      eikr(i)=eikx(i,kx)*eiky(i,ky)*eikz(i,kz)

```

```

        sum(kcount)=sum(kcount)+ch(i)*eikr(i)
    enddo
enddo

call MPI_BARRIER( MPI_COMM_WORLD, ierr )
call MPI_ALLREDUCE( sum, sum_all, totk,&
& MPI_DOUBLE_COMPLEX, MPI_SUM,&
& MPI_COMM_WORLD, ierr )

do kcount=1,totk !maxk
    kx=kx_field(kcount)
    ky=ky_field(kcount)
    kz=kz_field(kcount)
    if(kx == 0) then
        fact=1.d0
    else
        fact=2.d0
    endif
    vcoulk=vcoulk+fact*kvec(kcount)*conjg(sum_all(kcount))*
sum_all(kcount)
enddo

end subroutine kwald

! K-Space part of the coulombic force by ewald method
! in : ch,pos,kvec      out: ckf

subroutine fkwald(posloc,kvec,ckf)

    use MYMPI
    use typeDefs
    use program_wide_constants
    use charge_and_mass,only: chsi,cho,chna,ch
    use rsum_commons,only: twopi,twopidlbox
    use ksum_commons,only: kmax,ksqmax,kx_field,ky_field,kz_fi
eld

    use ewald,only: totk

    implicit none

    real(DOUBLE) :: posloc(partdimloc),ckf(partdimloc)
    real(DOUBLE) :: kvec(maxk)
    complex(DOUBLE) :: eikx(1:nchpartloc,0:kmax_para)
    complex(DOUBLE) :: eiky(1:nchpartloc,-kmax_para:kmax_para)
    complex(DOUBLE) :: eikz(1:nchpartloc,-kmax_para:kmax_para)
    complex(DOUBLE) :: tsum(maxk),tsum_all(maxk)

    integer(DEFAULT) :: i,j
    integer(DEFAULT) :: kx,ky,kz,kcount
    real(DOUBLE) :: tfi,kxd,kyd,kzd,kvec,factor
    complex(DOUBLE) :: tsum2,tmp,eikri,tforce

    ! construct exp(ikr) for all ions and k-vectors
    ! calculate kx, ky, kz = 0, -1 and 1 explicitly

    do i=1,nlocal
        eikx(i,0)=(1.d0,0.d0)
        eiky(i,0)=(1.d0,0.d0)
        eikz(i,0)=(1.d0,0.d0)
        eikx(i,1)=cmplx(cos(twopidlbox*posloc(3*i-2)),&
& sin(twopidlbox*posloc(3*i-2)))
        eiky(i,1)=cmplx(cos(twopidlbox*posloc(3*i-1)),&
& sin(twopidlbox*posloc(3*i-1)))
        eikz(i,1)=cmplx(cos(twopidlbox*posloc(3*i)),&
& sin(twopidlbox*posloc(3*i)))
        eiky(i,-1)=conjg(eiky(i,1))
        eikz(i,-1)=conjg(eikz(i,1))
    enddo

    ! calculate remaining kx, ky and kz by recurrence

    do j=2,kmax
        do i=1,nlocal
            eikx(i,j)=eikx(i,j-1)*eikx(i,1)
            eiky(i,j)=eiky(i,j-1)*eiky(i,1)
            eiky(i,-j)=conjg(eiky(i,j))
            eikz(i,j)=eikz(i,j-1)*eikz(i,1)
            eikz(i,-j)=conjg(eikz(i,j))
        enddo
    enddo

    ! now calculate sum_i exp(ik.r_i)*ch_i for every k-vecto
r and
    ! write the result into tsum_all(*)
    !
    do kcount=1,totk !maxk

        kx=kx_field(kcount)

```

```

        ky=ky_field(kcount)
        kz=kz_field(kcount)

        tsum(kcount)=(0.d0,0.d0)
        do j=1,nlocal
            tsum(kcount)=tsum(kcount)+eikx(j,kx)*eiky(j,ky)*eikz
(j,kz)*ch(j)
        enddo

    enddo

    call MPI_BARRIER( MPI_COMM_WORLD, ierr )
    call MPI_ALLREDUCE( tsum, tsum_all, totk,MPI_DOUBLE_COMPLE
X, MPI_SUM,MPI_COMM_WORLD, ierr )

    ! after initiating ckf(*) calculate the whole k space
    ! Ewald sum and write the result into ckf(*)

    do j=1,3*nlocal; ckf(j)=0.d0; enddo

    do kcount=1,totk !maxk

        kx=kx_field(kcount)
        ky=ky_field(kcount)
        kz=kz_field(kcount)

        if(kx == 0) then
            factor=1.d0
        else
            factor=2.d0
        endif

        kvec=kvec(kcount)
        kxd=kx
        kyd=ky
        kzd=kz
        tforce=factor*kvec*tsum_all(kcount)
        do i=1,nlocal
            eikri=conjg(eikx(i,kx)*eiky(i,ky)*eikz(i,kz))
            tfi=aimag(eikri*tforce*ch(i))
            ckf(3*i-2)=ckf(3*i-2)+tfi*kxd
            ckf(3*i-1)=ckf(3*i-1)+tfi*kyd
            ckf(3*i)=ckf(3*i)+tfi*kzd
        enddo

    enddo ! kcount

    do i=1,3*nlocal; ckf(i)=-2.d0*ckf(i)*twopi; enddo

end subroutine fkwald

!
! This routine does the calculation of the k-space part of t
he energy and the force.
! Since the first two loops with calculations are the same f
or both routines, and
! much time is spent calculating these, and since force need
s both, k-space energy and
! force, this routine speeds up the sim.
!
subroutine f_kwald(posloc,kvec,ckf,vcoulk)

    use MYMPI
    use typeDefs
    use program_wide_constants
    use charge_and_mass,only: chsi,cho,chna,ch
    use rsum_commons,only: twopi,twopidlbox
    use ksum_commons,only: kmax,ksqmax,kx_field,ky_field,kz_fi
eld

    use ewald,only: totk

    implicit none

    real(DOUBLE) :: posloc(partdimloc),ckf(partdimloc)
    real(DOUBLE) :: kvec(maxk)
    complex(DOUBLE) :: eikx(1:nchpartloc,0:kmax_para)
    complex(DOUBLE) :: eiky(1:nchpartloc,-kmax_para:kmax_para)
    complex(DOUBLE) :: eikz(1:nchpartloc,-kmax_para:kmax_para)
    complex(DOUBLE) :: sum(maxk),sum_all(maxk) ! sum of fkwa
ld

    integer(DEFAULT) :: i,j,kx,ky,kz,kcount
    real(DOUBLE) :: vcoulk
    real(DOUBLE) :: tfi,factor
    complex(DOUBLE) :: eikri,tforce

    ! construct exp(ikr) for all ions and k-vectors
    ! calculate kx, ky, kz = 0, -1 and 1 explicitly

    do i=1,nlocal

```

```

    eikx(i,0)=(1.d0,0.d0)
    eiky(i,0)=(1.d0,0.d0)
    eikz(i,0)=(1.d0,0.d0)
    eikx(i,1)=cplx(cos(twopidlbox*posloc(3*i-2)),sin(twopi
dlbox*posloc(3*i-2)))
    eiky(i,1)=cplx(cos(twopidlbox*posloc(3*i-1)),sin(twopi
dlbox*posloc(3*i-1)))
    eikz(i,1)=cplx(cos(twopidlbox*posloc(3*i)) ,sin(twopi
dlbox*posloc(3*i)))
    eiky(i,-1)=conjg(eiky(i,1))
    eikz(i,-1)=conjg(eikz(i,1))
    enddo

! calculate remaining kx, ky and kz by recurrence

do j=2,kmax
do i=1,nlocal
    eikx(i,j)=eikx(i,j-1)*eikx(i,1)
    eiky(i,j)=eiky(i,j-1)*eiky(i,1)
    eiky(i,-j)=conjg(eiky(i,j))
    eikz(i,j)=eikz(i,j-1)*eikz(i,1)
    eikz(i,-j)=conjg(eikz(i,j))
enddo
enddo

! now calculate sum_i exp(ik.r_i)*ch_i for every k-vecto
r and
! write the result into tsum_all(*)
!
do kcount=1,totk
    kx=kx_field(kcount)
    ky=ky_field(kcount)
    kz=kz_field(kcount)
    sum(kcount)=(0.d0,0.d0)
    do i=1,nlocal
        sum(kcount)=sum(kcount)+ch(i)*eikx(i,kx)*eiky(i,ky)*
eikz(i,kz)
    enddo
enddo

call MPI_BARRIER( MPI_COMM_WORLD, ierr )
call MPI_ALLREDUCE( sum, sum_all, totk,MPI_DOUBLE_COMPLEX,
MPI_SUM,MPI_COMM_WORLD, ierr )

! after initiating ckf(*) calculate the whole k space
! Ewald sum and write the result into ckf(*)

do j=1,3*nlocal
    ckf(j)=0.d0
enddo

vcoulk=0.d0

do kcount=1,totk !maxk

    kx=kx_field(kcount)
    ky=ky_field(kcount)
    kz=kz_field(kcount)

    if(kx == 0) then
        factor=1.d0
    else
        factor=2.d0
    endif

    vcoulk=vcoulk+factor*kvec(kcount)*conjg(sum_all(kcount)
)*sum_all(kcount) ! kwald done

    tforce=factor*kvec(kcount)*sum_all(kcount)
    do i=1,nlocal
        eikri=conjg(eikx(i,kx)*eiky(i,ky)*eikz(i,kz))
        tfi=aimag(eikri*tforce*ch(i))
        ckf(3*i-2)=ckf(3*i-2)+tfi*kx
        ckf(3*i-1)=ckf(3*i-1)+tfi*ky
        ckf(3*i) =ckf(3*i) +tfi*kz
    enddo

enddo ! kcount

do i=1,3*nlocal; ckf(i)=-2.d0*ckf(i)*twopi; enddo

end subroutine f_kwald

! This subroutine adds the number 'number' to the string 'st
ring'

subroutine addnumtostring(string,number)
    use typeDefs

    implicit none

    integer(DEFAULT) :: i, strlen, number, nodig, num, snum
    character(*) string

    snum=number
    do i=len(string),1,-1
        if(string(i:i) /= ' ') goto 10
    enddo
    10 strlen=i

    nodig=int(log10(1.0*snum+0.1))+1
    do i=nodig,1,-1
        num=snum/10**(i-1)
        string(strlen+1:strlen+1)=char(48+num)
        strlen=strlen+1
        snum=snum-num*10**(i-1)
    enddo

end subroutine addnumtostring

```





# Literaturverzeichnis

- [1] R. Zallen. *The Physics of Amorphous Solids*. Wiley, New York, 1983.
- [2] J. Jäckle. Models of the glass transition. *Rep. Prog. Phys.*, 49:171, 1986.
- [3] R. Zarzycki, editor. *Materials Science and Technology*, volume 9. VCH, Weinheim, 1991.
- [4] U. Mohanty. Supercooled liquids. *Adv. Chem. Phys.*, 89:89, 1995.
- [5] glassonline. URL <http://www.glassonline.com/infoserv/history.html>, 28. May 2004.
- [6] J. Horbach. *Molekulardynamiksimulationen zum Glasübergang von Silikatschmelzen*. PhD thesis, Johannes Gutenberg–Universität Mainz, 1998.
- [7] M. Tomozawa and M. Yoshiyagawa. *Glastech. Ber.*, 56:939, 1983.
- [8] W. Vogel. *Glaschemie*. Springer, 1992.
- [9] G. N. Greaves. Structural studies of the mixed alkali effect in disilicate glasses. *Sol. State Ion.*, 105:243–248, 1998.
- [10] W. Smith, G. N. Greaves, and M. J. Gillian. Computer–simulation of sodium disilicate glass. *J. Chem. Phys.*, 103(8):3091–3097, 1995.
- [11] W. Smith, G. N. Greaves, and M. J. Gillian. The structure and dynamics of sodium disilicate glass by molecular-dynamics simulation. *J. Non-Cryst. Sol.*, 193:267–271, 1995.
- [12] G. N. Greaves, W. Smith, E. Giolotto, and E. Pantos. Local structure, microstructure and glass properties. *J. Non-Cryst. Sol.*, 222:13–24, 1997.
- [13] J. Oviedo and J. F. Sanz. Molecular-dynamics simulations of  $(\text{Na}_2\text{O})_x(\text{SiO}_2)_{1-x}$  glasses: Relation between distribution and diffusive behavior of Na atoms. *Phys. Rev. B*, 58(14):9047–9053, 1998.
- [14] J. Horbach, W. Kob, and K. Binder. Structural and dynamical properties of sodium silica melts: An investigation by molecular dynamics computer simulation. *Chem. Geol.*, 174: 87–101, 2001.
- [15] J. Horbach and W. Kob. The structure and dynamics of sodium disilicate. *Phil. Mag. B*, 79:1981, 1999.

- [16] J. Horbach, W. Kob, and K. Binder. Dynamics of Sodium in Sodium Disilicate: Channel Relaxation and Sodium Diffusion. *Phys. Rev. Lett.*, 88(12):125502, 2002.
- [17] A. Winkler. *Molekulardynamik–Untersuchungen zur atomistischen Struktur & Dynamik von binären Mischgläsern  $(\text{Na}_2\text{O})(x \cdot \text{SiO}_2)$  und  $(\text{Al}_2\text{O}_3)(2 \cdot \text{SiO}_2)$* . PhD thesis, Johannes Gutenberg–Universität Mainz, Mai 2002.
- [18] E. Sunyer, P. Jund, W. Kob, and R. Jullien. Molecular dynamics study of the diffusion of sodium in amorphous silica. *J. Non–Cryst. Sol.*, 307:939–945, 2002.
- [19] E. Sunyer, P. Jund, and R. Jullien. Characterization of channel diffusion in a sodium tetrasilicate glass via molecular-dynamics simulations. *Phys. Rev. B*, 65:214203, 2002.
- [20] E. Sunyer, P. Jund, and R. Jullien. Matrix-controlled channel diffusion of sodium in amorphous silica. *J. Phys.: Condens. Matter*, 15:L431–L437, 2003.
- [21] E. Sunyer, P. Jund, and R. Jullien. Numerical investigation of ionic transport in glasses: the example of sodium in amorphous silica. *J. Phys.: Condens. Matter*, 15:S1659–S1671, 2003.
- [22] R. D. Banhatti and A. Heuer. Structure and dynamics of lithium silicate melts: molecular dynamics simulations. *Phys. Chem. Chem. Phys.*, 3:5104–5108, 2001.
- [23] H. Lammert, M. Kunow, and A. Heuer. Complete identification of alkali sites in ion conducting lithium silicate glasses: A computer study of ion dynamics. *Phys. Rev. Lett.*, 90(21):215901, 2003.
- [24] H. Lammert and A. Heuer. Emergence of conduction channels in lithium silicate. *Phys. Rev. B*, 70(2):024204, 2004.
- [25] H. Doweidar. The density of alkali silicate glasses in relation to the microstructure. *J. Non-Cryst. Sol.*, 194:155–162, 1996.
- [26] S. Balasubramanian and K. J. Rao. Preferential Paths in Alkali Ion Migration and the Mixed Alkali Effect in Silicate Glasses. *J. Phys. Chem.*, 97(35):8835–8838, 1993.
- [27] S. Balasubramanian and K. J. Rao. A molecular dynamics study of the mixed alkali effect in silicate glasses. *J. Non-Cryst. Sol.*, 181:157–174, 1995.
- [28] A. A. Tesar and A. K. Varshneya. Molecular dynamics simulation of alkali-silicate glass structures. *J. Chem. Phys.*, 87:2986, 1987.
- [29] J. Swenson and S. Adams. Mixed alkali effect in glasses. *Phys. Rev. Lett.*, 90(15):155507–1, 2003.
- [30] J. Habasaki and Y. Hiwatari. Loosening of the structure in a mixed alkali glass. *Phys. Rev. E*, 58(4):5111–5114, 1998.
- [31] A. N. Cormack and Yuan Cao. *Modelling of Minerals and Silicated Materials*, chapter Molecular Dynamics Simulation of Silicate, pages 227–271. Kluwer Academic Publishers, Netherlands, 1997.

- [32] T. F. Soules. A molecular dynamic calculation of the structure of sodium silicate glasses. *J. Chem. Phys.*, 71(11):4571–4578, 1979.
- [33] M. Kunow and A. Heuer. A simple harmonic model as a caricature for mismatch and relaxation effects for ion hopping dynamics in solid electrolytes. *J. Chem. Phys.*, 119(4):2338–2347, 2003.
- [34] X. Yuan and A. N. Cormack. Si–O–Si bond angle and torsion angle distribution in vitreous silica and sodium silicate glasses. *J. Non-Cryst. Sol.*, 319:31–43, 2003.
- [35] F. Kargl. Daten zum elastischen Strukturfaktor aus Neutronenstreuungsmessungen von Lithiumdisilikat für  $T = 1205\text{ K}, 1405\text{ K}, 1605\text{ K}$  / Private Mitteilung / Technische Universität München.
- [36] R. L. Myuller. *Zr. Tek. Fiz.*, 25:236, 1955.
- [37] J. M. Stevels. *Handbuch der Physik*, volume 20/2, page 350. Springer, Berlin, 1957.
- [38] C. Huang and A. N. Cormack. Structure and energetics in mixed-alkali-metal silicate glasses from molecular dynamics. *J. Mater. Chem.*, 2:281, 1992.
- [39] K. Funke. Jump relaxation in solid electrolytes. *Prog. Sol. Stat. Chem.*, 22:111, 1993.
- [40] K. Funke. Jump relaxation model and coupling model - a comparison. *J. Non-Cryst. Sol.*, 172–174:1215, 1994.
- [41] P. Maass, A. Bunde, and M. D. Ingram. Ion transport anomalies in glasses. *Phys. Rev. Lett.*, 68:3064, 1992.
- [42] A. Bunde, M. D. Ingram, and P. Maass. The dynamic structure model for ion-transport in glasses. *J. Non-Cryst. Solids*, 172:1222, 1994.
- [43] P. Maass. Towards a theory for the mixed alkali effect in glasses. *J. Non-Cryst. Sol.*, 225:35–46, 1999.
- [44] J. E. Davidson, M. D. Ingram, A. Bunde, and K. Funke. Ion hopping processes and structural relaxation in glassy materials. *J. Non-Cryst. Sol.*, 203:246, 1996.
- [45] R. Kirchheim. The mixed alkali effect as a consequence of network density and site energy distribution. *J. Non. Cryst. Sol.*, 272:85–102, 2000.
- [46] C. Walker and S. Morton. Surface Science Techniques.  
URL <http://www.uksaf.org/tech/exafs.html>, 10. September 2004.
- [47] G. N. Greaves, S. J. Gurman, C. R. A. Catlow, S. Houde-Walter, A. V. Chadwick, C. M. B. Henderson, and B. R. Dobson. *Philos. Mag. A*, 64:1059, 1991.
- [48] S. N. Houde-Walter, J. M. Inman, A. J. Dent, and G. N. Greaves. Sodium and silver environments and ion-exchange processes in silicate and aluminosilicate glasses. *J. Phys. Chem.*, 97:9330, 1993.

- [49] U. Bengtzelius, W. Götze, and A. Sjölander. *J. Phys. C*, 17:5915, 1984.
- [50] W. Kob. The Mode–Coupling Theory of the Glass Transition. In J. Fourkas, D. Kivelson, U. Mohanty, and K. Nelson, editors, *Experimental and Theoretical Approaches to Supercooled Liquids: Advances and Novel Applications*. ACS Books, Washington, 1997.
- [51] J. Horbach and W. Kob. The structural relaxation of molten sodium disilicate. *J. Phys.: Condens. Matter*, 14:9237–9253, 2002.
- [52] L. Sjögren. Kinetic theory of current fluctuations in simple classical liquids. *Phys. Rev. A*, 22:2866, 1980.
- [53] R. Schilling. Theories of the Structural Glass Transition. In G. Radons, W. Just, and P. Häussler, editors, *Collective Dynamics of Nonlinear and Disordered Systems*. Springer, Berlin, 2003.
- [54] Theme Issue on Relaxation Kinetics in Supercooled Liquids–Mode Coupling Theory and its Experimental Tests. *Transport Theory and Statistical Physics*, 24(6–8), 1995.
- [55] R. Schilling. *Disorder Effects on Relaxational Processes*, page 193. Springer, Berlin, 1994.
- [56] H. Z. Cummins, G. Li, W. M. Du, and J. Hernandez. Relaxational dynamics in supercooled liquids: experimental tests of the mode coupling theory. *Physica A*, 204:169, 1994.
- [57] W. Götze. Liquids, Freezing and the Glass Transition. Les Houches – Session LI, 1989.
- [58] W. Götze and L. Sjögren. Relaxation processes in supercooled liquids. *Rep. Prog. Phys.*, 55:241, 1992.
- [59] J. Horbach. Private Mitteilung / Johannes Gutenberg–Universität Mainz, 2004.
- [60] T. Franosch, M. Fuchs, W. Götze, M. R. Mayr, and A. P. Singh. Asymptotic laws and preasymptotic correction formulas for the relaxation near glass–transition singularities. *Phys. Rev. E*, 55(6):7153, 1997.
- [61] G. Li, W. M. Du, X. K. Chen, H. Z. Cummins, and N. J. Tao. Testing mode-coupling predictions for alpha and beta relaxation in  $Ca_{0.4}K_{0.6}(NO_3)_{1.4}$  near the liquid-glass transition by light scattering. *Phys. Rev. A*, 45:3867, 1992.
- [62] W. van Mengen and S. M. Underwood. Glass transition in colloidal hard spheres: Measurement and mode–coupling–theory analysis of the coherent intermediate scattering function. *Phys. Rev. E*, 49:4206, 1994.
- [63] T. Voigtmann. Modenkopplungsrechnungen zu Natriumdisilikat auf Grundlage von Simulationsdaten / Private Mitteilung / University of Edinburgh.
- [64] M. Nauroth and W. Kob. Quantitative test of the mode-coupling theory of the ideal glass transition for a binary lennard–jones system. *Phys. Rev. E*, 55:657, 1997.

- [65] F. Sciortino and W. Kob. Debye-Waller Factor of Liquid Silica: Theory and Simulation. *Phys. Rev. Lett.*, 86:648, 2001.
- [66] J. Horbach, W. Kob, and K. Binder. The dynamics of melts containing mobile ions: computer simulations of sodium silicates. *J. Phys.: Condens. Matter*, 15:S903, 2003.
- [67] H. Kuchling. *Taschenbuch der Physik*. Fachbuchverlag, Leipzig; Köln, 15. verb. edition, 1995.
- [68] URL <http://www.physik.rwth-aachen.de/group/IIIphys/INFOS/Exscript/3Kapitel/III5Kapitel.html>, 28. Oktober 2004.
- [69] D. Herzbach. *Comparison of model potentials for molecular dynamics simulation of crystalline silica*. PhD thesis, Johannes Gutenberg–Universität Mainz, Juni 2004.
- [70] S. Tsuneyuki, M. Tukada, M. Aoki, and Y. Matsui. First–Principles Interatomic Potential of Silica Applied to Molecular Dynamics. *Phys. Rev. Lett.*, 61:869, 1988.
- [71] J. Habasaki and I. Okada. Molecular dynamics simulation of alkali silicates based on the quantum mechanical potential surfaces. *Molecular Simulation*, 9(5):319, 1992.
- [72] K. Vollmayr. *Abkühlratenabhängigkeit von strukturellen Gläsern: Eine Computersimulation*. PhD thesis, Johannes–Gutenberg Universität Mainz, November 1995.
- [73] C. Mischler. *Molekulardynamik–Simulationen zur Struktur von SiO<sub>2</sub>–Oberflächen mit adsorbiertem Wasser*. PhD thesis, Johannes Gutenberg–Universität Mainz, 2002.
- [74] M. Winter. Webelements<sup>TM</sup> Periodic table (professional edition). URL <http://www.webelements.com/>, 24. Oktober 2000.
- [75] H. C. Andersen. Molecular dynamics simulations at constant pressure and/or temperature. *J. Chem. Phys.*, 72:2384, 1980.
- [76] J. R. Fox and H. C. Andersen. Molecular dynamics simulations of a supercooled monatomic liquid and glass. *J. Phys. Chem.*, 88:4019, 1984.
- [77] M. P. Allen and D. Tildesley. *Computer Simulation of Liquids*. Oxford University Press, 1997.
- [78] K. Binder and D. W. Heermann. *Monte Carlo simulation in statistical physics*. Springer Verlag, Berlin, Heidelberg, New York, 3. edition, 1997.
- [79] J. Horbach, W. Kob, K. Binder, and C. A. Angell. Finite size effects in simulations of glass dynamics. *Phys. Rev. E*, 54(6):R5897–R5900, 1996.
- [80] E. Madelung. Das elektrische Feld in Systemen von regelmässig angeordneten Punktladungen. *Phys. Z.*, 19:524–532, 1918.
- [81] S. G. Brush, H. L. Sahlin, and E. Teller. Monte Carlo Study of a One-Component Plasma. *J. Chem. Phys.*, 45:2102, 1966.

- [82] J. P. Hansen, I. R. McDonald, and P. Vieillefosse. Statistical mechanics of dense ionized matter. VIII. Dynamical properties of binary ionic mixtures.
- [83] D. Wolf, P. Keblinski, S. R. Phillpot, and J. Eggebrecht. Exact method for the simulation of Coulombic systems by spherically truncated, pairwise r<sup>-1</sup> summation. *J. Chem. Phys.*, 110:7980, 2001.
- [84] P. Demontis, S. Spanu, and G. B. Suffriti. Application of the Wolf method for the evaluation of Coulombic interactions to complex condensed matter systems: Aluminosilicates and water. *J. Chem. Phys.*, 114:7980, 2001.
- [85] P. P. Ewald. *Annalen der Physik*, 64:253, 1921.
- [86] M. J. Lighthill. *Einführung in die Fourier–Analysis*. Bibliographisches Institut, Mannheim, 1966.
- [87] M. T. Dove. *Introduction to Lattice Dynamics*. Cambridge University Press, Cambridge, 1993.
- [88] D. Frenkel and B. Smit. *Understanding Molecular Simulation*. Academic Press, London, 2002.
- [89] M. Abramovitz and I. A. Stegun. *Handbook of Mathematical Functions*. Dover Publications, New York, 1972.
- [90] J. Horbach and W. Kob. Static and dynamic properties of a viscous silica melt. *Phys. Rev. B*, 60:3169–3181, 1999.
- [91] J. Horbach and W. Kob. Relaxation dynamics of a viscous silica melt: The intermediate scattering functions. *Phys. Rev. E*, 64:041503, 2001.
- [92] P. Jund and R. Jullien. Computer investigation of the energy landscape of amorphous silica. *Phys. Rev. Lett.*, 83:2210–2213, 1999.
- [93] P. Jund and R. Jullien. Molecular–dynamics calculation of the thermal conductivity of vitreous silica. *Phys. Rev. B*, 59:13707–13711, 1999.
- [94] P. Jund and R. Jullien. Numerical study of the structural and thermal properties of vitreous silica. *Mol. Simulat.*, 24:25–49, 2000.
- [95] M. Rarivomanantsoa, P. Jund, and R. Jullien. Classical molecular dynamics simulations of amorphous silica surfaces. *J. Phys. Condens. Mat.*, 13(31):6707–6718, 2001.
- [96] S. N. Taraskin and S. R. Elliot. Nature of vibrational excitations in vitreous silica. *Phys. Rev. B*, 56:8605–8622, 1997.
- [97] S. N. Taraskin and S. R. Elliot. Anharmonicity and localization of atomic vibrations in vitreous silica. *Phys. Rev. B*, 59:8572–8585, 1999.
- [98] S. N. Taraskin and S. R. Elliot. Propagation of plane–wave vibrational excitations in disordered systems. *Phys. Rev. B*, 61:12017–12030, 2000.

- [99] S. N. Taraskin and S. R. Elliot. Ioffe–regel crossover for plane–wave vibrational excitations in vitreous silica. *Phys. Rev. B*, 61:12031–12037, 2000.
- [100] K. Vollmayr, W. Kob, and K. Binder. Cooling rate effects in amorphous silica: A computer simulation study. *Phys. Rev. B*, 54:15808, 1996.
- [101] B. W. H. van Beest, G. J. Kramer, and R. A. van Santen. Force fields for silicas and aluminophosphates based on ab initio calculations. *Phys. Rev. Lett.*, 64:1955, 1990.
- [102] G. J. Kramer, A. J. de Man, and R. A. van Santen. Zeolites versus aluminosilicate clusters: the validity of a local description. *J. Am. Chem. Soc.*, 113:6435, 1991.
- [103] R. Car and M. Parinello. Unified Approach for Molecular Dynamics and Density–Functional Theory. *Phys. Rev. Lett.*, 55:2471, 1985.
- [104] J. Habasaki and I. Okada. Origins of the two-step relaxation and the boson peak in an alkali silicate glass studied by molecular–dynamics simulation. *Phys. Rev. E*, 52(3): 2681–2687, 1995.
- [105] J. Habasaki and Y. Hiwatari. Characteristics of slow and fast ion dynamics in a lithium metasilicate glass. *Phys. Rev. E*, 59(6):6962–6966, 1999.
- [106] A. Heuer, M. Kunow, M. Vogel, and R. D. Banhatti. Characterization of the complex ion dynamics in lithium silicate glasses via computer simulations. *Phys. Chem. Chem. Phys.*, 4:3185–3192, 2002.
- [107] O. V. Mazurin, M. V. Streltsina, and T. P. Shvaiko-Shvaikovskaya. *Handbook of glass data, Part A: Silica Glass and Binary Silicate Glasses*. Elsevier, Amsterdam, 1983.
- [108] The Message Passing Interface (MPI) standard.  
URL <http://www-unix.mcs.anl.gov/mpi/>, 13. September 2004.
- [109] R. Klein. *Statistische Mechanik*. Fakultät für Physik, Dekanat, Universität Konstanz, 1 edition, November 1997.
- [110] U. Balucani and M. Zoppi. *Dynamics of the Liquid State*. Oxford University Press Inc., New York, 1994.
- [111] A. Roder. *Molekulardynamik–Simulationen zu Oberflächeneigenschaften von Siliziumdioxid–Schmelzen*. PhD thesis, Johannes–Gutenberg Universität Mainz, 2000.
- [112] J. P. Rino, I. Ebbsjö, R. K. Kalia, A. Nakano, and P. Vashishta. Structure of rings in vitreous  $\text{SiO}_2$ . *Phys. Rev. B*, 47:3053, 1993.
- [113] F. L. Galeener. Planar rings in glasses. *Solid State Commun.*, 44:1037, 1982.
- [114] A. Meyer, J. Horbach, W. Kob, F. Kargl, and H. Schober. Channel Formation and Intermediate Range Order in Sodium Silicate Melts and Glasses. *Phys. Rev. E*, 93(2): 027801, 2004.



- [115] A. Meyer, H. Schober, and D. B. Dingwell. Structure, structural relaxation and ion diffusion in sodium disilicate melts. *Europhys. Lett.*, 59(5):708–713, 2002.
- [116] J. Neufeind. On the partial structure factors of molten zinc chloride. *Chem. Phys.*, 3: 3987–3993, 2001.
- [117] Special feature section of neutron scattering lengths and cross sections of the elements and their isotopes. *Neutron News*, 3(3):29–37, 1992.
- [118] J. M. Carpenter and C. A. Pelizzari. Inelastic neutron scattering from amorphous solids. II. Interpretation of measurements. *Phys. Rev. B.*, 12:2397, 1975.
- [119] S. W. Lovesey. *Theory of Neutron Scattering from Condensed Matter*, volume 1. Clarendon Press, Oxford, 1986.
- [120] Y. Rosenfeld and P. Tarazona. Density functional theory and the asymptotic high density expansion of the free energy of classical solids and fluids. *Mol. Phys.*, 95:141–150, 1998.
- [121] L. Shartsis, S. Spinner, and W. Capps. *J. Am. Ceram. Soc.*, 35:155–160, 1952.
- [122] J. C. Young, F. W. Glaze, C. A. Faick, and A. N. Finn. *J. Res. Nat. Bur. Stand.*, 22: 453–464, 1939.
- [123] J. O’M Bockris, J. W. Tomlinson, and J. L. White. *Trans. Faraday Soc.*, 52(3):299–310, 1956.
- [124] R. L. Mozzi and B. E. Warren. The Structure of Vitreous Silica. *J. Appl. Cryst.*, 2: 164–172, 1969.
- [125] W. Humphrey, A. Dalke, and K. Schulten. Vmd – Visual Molecular Dynamics. *J. Molec. Graphics*, 14:33–38, 1996. URL <http://www.ks.uiuc.edu/Research/vmd/>.
- [126] F. Kargl, A. Meyer, M. M. Koza, and H. Schober. submitted to *Phys. Rev. B*, 2004.
- [127] R. Knoche, D. B. Dingwell, F. A. Seifert, and S. L. Webb. Non-linear properties of supercooled liquids in the system  $\text{Na}_2\text{O} - \text{SiO}_2$ . *Chem. Geol.*, 116:1–16, 1994.
- [128] Y. P. Gupta and T. B. King. *Transactions of the Metallurgical Society of AIME*, 239: 1701–1707, 1967.
- [129] J. R. Johnson, R. H. Bristow, and J. Blau. *J. Am. Ceram. Soc.*, 34:165, 1951.
- [130] J. P. Hansen and I. R. McDonald. *Theory of Simple Liquids*. Academic Press, London, 1986.
- [131] S. Glasstone, K. J. Laidler, and H. Eyring. *The Theory of Rate Processes*. McGraw-Hill, New York, 1941.
- [132] S. K. Das, J. Horbach, and K. Binder. Transport phenomena and microscopic structure in partially miscible binary fluids: A simulation study of the symmetrical Lennard-Jones mixture. *J. Chem. Phys.*, 119:1547, 2003.

- [133] J. L. Souquet and M. Lèvy. A single microscopic approach for ionic transport in glassy and polymer electrolytes. *Sol. Stat. Ion.*, 70–71:337, 1994.
- [134] R. Terai. The mixed alkali effect in the  $\text{Na}_2\text{O}\text{--}\text{Cs}_2\text{O}\text{--}\text{SiO}_2$  glasses. *J. Non-Cryst. Sol.*, 6:121, 1971.
- [135] A. Jünger. Mathematische Modellierung mit Differentialgleichungen, 2003. unkorrigiertes Vorlesungsskript, Fachbereich Mathematik und Informatik, Johannes Gutenberg–Universität Mainz.