

## 3 Löslichkeitsberechnung auf Basis einer verallgemeinerten DPD Methode

### 3.1 Generalisierung der DPD Methode

#### 3.1.1 Ziele der Generalisierung

Um die von Boulougouris und Theodorou entwickelte DPD Methode zur Berechnung von chemischem Exzesspotenzial und Löslichkeitskoeffizienten für ein wesentlich breiteres Materialspektrum zugänglich zu machen, sind Erweiterungen notwendig. Da die bisherige Implementierung durch Boulougouris sehr speziell an die untersuchten, einfachen Modellsysteme angepasst war, konnte nicht auf die vorhandenen Software aufgebaut werden, sondern die Rechenmethode wurde mit völlig neu geschriebener Software realisiert. Ziele waren dabei

- Die Methode soll mit (nahezu) beliebigen Polymeren als Matrix arbeiten können. Mit anderen Worten: Die Matrix-Teilchen bestehen aus (viel) mehr als *einem* Wechselwirkungs-Zentrum (Atom).
- Die Methode soll Penetranten mit einer atomaren Eigenstruktur jenseits der simplen Kugelgestalt verarbeiten können. Mit anderen Worten: der explizite atomare Aufbau z. B. eines Benzol- oder Heptan-Moleküls soll berücksichtigt werden — dies ist (im Gegensatz zu Sauerstoff oder Methan) nicht mit einer simplen Kugel adäquat realisierbar, da die äußere Gestalt der genannten Moleküle eher linsen- bzw. wurstförmig ist.
- Verwendung von detailliert-atomistischen Simulationen der untersuchten Systeme, so dass die Wechselwirkung quasi beliebiger Moleküle untersucht werden kann, soweit sinnvolle Kraftfeld-Parameter verfügbar sind. (Also keine Beschränkung mehr auf Lennard-Jones Kugeln oder *united atoms* Repräsentationen.)

- Die zu entwickelnde Software sollte vergleichsweise einfach an unterschiedliche (MC oder MD) Simulations-Programme adaptierbar sein, und auf unterschiedlichen Plattformen laufen können.

### 3.1.2 Polymere als Matrix

Ein erster neuer Aspekt in dieser Arbeit ist die Übertragung der DPD Methode auf Polymere, indem diese als (hochviskose) Lösungsmittel-Komponente (Matrix) betrachtet werden. Die notwendigen Anpassungen der Theorie sind eher gering und beschränken sich vor allem auf die Umdeutung der Variablen der DPD Gleichung (23). Wenn man von ein-atomigen Flüssigkeiten als Matrix über mehr-atomige Moleküle bis hin zu langkettigen Polymeren als Matrix geht, so betrifft dies vor allem die Bedeutung von  $N_1$ : Dieser Parameter zählt nun nicht mehr alle Matrix-Atome/-Moleküle, sondern stellt die Summe aller Atome der Matrix dar — unabhängig davon, ob es sich dabei dann um „freie“ Atome (wie bei flüssigem Argon), oder alle Rückgrat- und Substituenten-Atome eines Polymers handelt. Die Unterscheidung zwischen einzelnen Atomen und komplexen Molekülen/Polymeren bleibt die Aufgabe der verwendeten Simulations Software — für die Auswertung der gespeicherten Konfigurationen bleiben alle Atome unverändert auf ihrem Platz — es ist also völlig belanglos, ob sich ein Atom nun an seiner aktuellen Position befindet, weil es durch zufällige, thermische Bewegungen dorthin gekommen ist oder aufgrund von bindenden Wechselwirkungen mit anderen Atomen.

Wissenschaftlich interessanter als die neue Bedeutung von  $N_1$  ist die Übertragung der DPD Methode aus dem Bereich der Flüssig-Dampf-Gleichgewichte, in dem sie Boulougouris entwickelt und angewendet hat, in den Bereich der Löslichkeiten kleiner und mittelgroßer Moleküle in Polymeren. Es muss untersucht werden, ob die Methode für eine polymere Matrix überhaupt sinnvolle Ergebnisse zu liefern vermag.

### 3.1.3 Moleküle als Penetrant mit Eigenstruktur

Die Berücksichtigung der Struktur des Penetranten erfolgt bei der Berechnung von  $H(r_{i,2})$  (vgl. Gleichung (22)) und bei der geometrischen Ermittlung des zugänglichen Volumens  $\phi$ . Offensichtlich findet in der gleichen Matrix-Konfiguration ein strukturierter Penetrant ein anderes zugängliches Volumen vor als ein kugelförmiger Penetrant. Die daraus resultierenden Konsequenzen für die Berechnung von  $H$  und  $\phi$  werden in den Abschnitten 3.1.5 bzw. 3.3.2 erläutert.

Boulougouris und Theodorou konnten zeigen<sup>36</sup> [62], dass es grundsätzlich möglich ist, die DPD Methode auch bei Penetranten mit einer komplexen Struktur anzuwenden: Das chemische Exzesspotenzial (und die daraus abgeleitete Sorptionswärme  $\Delta H_S$ ) eines vereinfachten (6-Zentren) Benzol Modells in sich selbst (als Flüssigkeit) und in *united atoms* Poly(ethylen) (mit 1 Kugel pro  $-\text{CH}_2-$  Einheit) wurden berechnet und stimmten (in Größenordnung und Vorzeichen) mit der real gemessenen Temperaturabhängigkeit der genannten Systeme überein [63]:

$$\begin{aligned}\Delta H_S^{\text{sim}} &= (-35.5 \pm 10) \text{ kJ/mol} && \text{für Benzol in Poly(ethylen)} \\ \Delta H_S^{\text{exp}} &= (-24.28 \dots -28.66) \text{ kJ/mol} && \text{abh. von der Molmasse des Polymers} \quad .\end{aligned}$$

Was bislang noch unberücksichtigt bleibt sind die zusätzlichen Freiheitsgrade (neben der Translation in der Packung), die ein strukturierter (nicht kugelförmiger) Penetrant hat. Die Freiheitsgrade der Rotation des kompletten Penetranten ließen sich noch allgemein implementieren, doch für die inneren Freiheitsgrade der Verformung des Penetranten (Bindungswinkel, Torsion, ect.) müssten spezielle Sonderbehandlungen eingeführt werden. Allerdings würde durch die Berücksichtigung beispielsweise der Rotationsfreiheitsgrade nicht unbedingt eine qualitative Verbesserung der Berechnung erreicht werden, denn die Partikel *Entfernungs* Methoden nutzen ja gerade die Tatsache, dass sich die lokale Matrixkonformation dem Penetranten angepasst hat. Ein realistischer Penetrant passt jedoch schon nach einer einfachen Drehung meist nicht mehr in den (durch seine eigene Entfernung entstandenen) Hohlraum hinein.

---

<sup>36</sup>Siehe Fußnote auf Seite 25.

Aus den genannten Gründen werden auch bei mehrzentrigen Penetranten nur die Translationsfreiheitsgrade zur Ermittlung des Anteils  $\phi$  an zugänglichem Volumen berücksichtigt. Praktisch werden die weiteren *konformellen* Freiheitsgrade des Penetranten in der DPD Methode „indirekt“ berücksichtigt, wenn auch nicht während der Untersuchung einer einzelnen Konfiguration: Die einzelnen in einer Trajektorie gespeicherten Konfigurationen beinhalten nicht nur die Matrix, sondern auch den Penetranten in unterschiedlichen Konformation, so dass bei der Mittelung über einige hundert bis tausende Frames auch ein repräsentativer Anteil der möglichen Konformationen des Penetranten berücksichtigt wird.

### 3.1.4 Detailliert-atomistische Modelle

Der Übergang von einfachen Kugelmodellen zu detailliert-atomistischen Modellen stellt keine grundsätzlich neue Herausforderung für die DPD Methode dar; ähnlich wie beim Übergang von kleinen Molekülen zu Polymeren und strukturierten Penetranten, steigt lediglich die Anzahl der zu berücksichtigenden Kugeln im System bei der geometrischen Analyse. Der Vorteil durch diese Verallgemeinerung liegt vielmehr darin, dass es dadurch praktisch ermöglicht wird, ein sehr breites Spektrum von Polymeren und darin gelösten Penetranten zu untersuchen. Beispielsweise mit der universelle MD-Simulations-Software *Discover* in Verbindung mit dem modernen COMPASS Kraftfeld [64], anstatt zunächst sinnvolle *united atoms* Repräsentationen von Polymer und Penetrant zu finden, und die passenden Kraftfeld-Parameter zu ermitteln.

Die praktische Auswirkung dieser Verallgemeinerung ist, dass nun über alle  $N_2$  Atome, die zum Penetrant-Molekül gehören, summiert wird.  $N_1$  ist der Index für alle einzelnen Atome der Matrix. (Es ist auch möglich, dass mehr als ein Penetrant „2“ im System simuliert wird; davon wird jedoch nur ein Molekül als *das Penetrant-Molekül*  $N_2$  selektiert. Die restlichen gelösten Moleküle werden ebenfalls als Lösungsmittel „(1)“ angesehen und können bei einer weiteren Auswertung als  $N_2$  spezifiziert werden.)

### 3.1.5 Bedeutung des Hartkugel-Durchmessers

In der bisherigen Darstellung der DPD Methode wird das Penetrant-Molekül zunächst durch *eine* harte Kugel mit Durchmesser  $d$  ersetzt. Dieser Durchmesser wurde dann nachfolgend verwendet, um auf eine möglich Überlappung ( $H=0$ ) zwischen der harten Kugel und den restlichen Matrix-Atomen zu prüfen. (Vergleich der Zentren-Abstände  $r_{i,2}$  mittels derer Orts-Vektoren  $|\vec{r}_i - \vec{r}_2|$  in Gleichung (22).)

Sobald der Penetrant eine atomare Struktur hat, muss für jedes Atom des Penetranten überprüft werden, ob es mit der Matrix überlappt; der Index „2“ ist daher keine Konstante, die das *eine* Penetrant-Partikel identifiziert, sondern ebenfalls eine Laufzahl („ $j$ “) über alle Atome des Penetranten. Zugleich ist auch der Hartkugeldurchmesser  $d$  keine fest gewählte Größe mehr, sondern repräsentiert eine Schar von Werten  $d_{i,j}$  für jede  $(i,j)$ -Paarung von  $i$  Matrix- und  $j$  Penetrant-Atomen [5]. Bei der Verwendung von detailliert-atomistischen Modellen besteht auch die Matrix üblicherweise aus unterschiedlichen Atomsorten, so dass  $i$  hier die unterschiedlichen Atomtypen der Matrix symbolisiert.

Ursprünglich wurde  $d$  aus der Paar-Abstands-Verteilung<sup>37</sup> ermittelt; wobei gezeigt wurde, dass der exakte Wert von  $d$  in gewissen Grenzen nur den Fehler der Berechnung, nicht jedoch den Ergebniswert selbst beeinflusst [5]. Die anschauliche Betrachtung der möglichen Überlappung zweier Kugeln nutzt allerdings deren Zentren-Abstand und vergleich mit der Summe der jeweiligen *Radien* der beiden Kugeln (und nicht mit dem Durchmesser *einer* der beiden Kugeln). In Gleichung (22) wurde insofern davon ausgegangen, dass die beiden Hartkugel-Repräsentationen der (eventuell) überlappenden Matrix- und Penetrant-Atome gleich groß sind — eine Annahme die für flüssiges Argon und eventuell noch für *united atoms* Modelle gilt, nicht jedoch für die angestrebten detailliert-atomistischen Modelle! Beim Übergang zur Schar  $d_{i,j}$  wird daher der minimal erlaubte Abstand (zunächst) ohne Zuhilfenahme der Paar-Abstands-Verteilung

---

<sup>37</sup>Der optimale Wert für  $d$  ist 10% größer als die dichteste Annäherung von Penetrant- und Matrix-Atom während der kompletten Simulation.

ermittelt: Jedem Matrix- und Penetrant-Atom wird ein Hartkugelradius zugewiesen (s.u.). Der Radius des Matrix-Atom und ein bestimmter Bruchteil  $\gamma_r$  des Radius des Penetranten werden summiert und ergeben  $d_{i,j}$  für den Vergleich der entsprechenden Atomsorten-Paarung aus  $i$  und  $j$ :

$$d_{i,j} = r_i(\text{Matrix}) + \gamma_r r_j(\text{Penetrant}) \quad (25)$$

**Der Penetrant-Radien Skalierungsfaktor**  $\gamma_r$  für die repräsentativen Hartkugelradien der Atome eines Penetrant-Moleküls ist somit nun ein wichtiger Parameter mit dem die Qualität der Berechnung verbessert werden kann — wird er geändert, so ändern sich entsprechend auch alle  $d_{i,j}$ -Werte in der weiteren DPD Auswertung. Sinnvolle Werte für  $\gamma_r$  wurden im Bereich zwischen etwa 0.5 und 1.2 gefunden, je nach untersuchtem System. Im Anwendungsteil dieser Arbeit findet eine systematische Untersuchung des Einflusses von  $\gamma_r$  auf die berechneten Löslichkeiten für eine Reihe von Polymer-Systemen statt. Es hat sich dabei gezeigt, dass die verlässlichsten Werte für die Löslichkeit erhalten werden, wenn in einem System etwa 70 bis 95% aller untersuchten Konfigurationen keine Hartkugel-Überlappung aufweisen (sog. *Hit-Frames*). Bei Wahl eines passenden  $\gamma_r$ -Wertes werden dann zum einen ausreichend viele Konfigurationen untersucht, um eine gute statistische Qualität der Berechnung zu erhalten, andererseits werden jedoch die unwahrscheinlichen Konfigurationen (mit zu starker Annäherung des Penetranten an die Matrix-Atome) und entsprechend abstoßender Wechselwirkungsenergie noch aussortiert. In der radiusabhängigen Darstellung („Radien-Plot“, siehe 4.2.2) der Ergebnisse dieser Arbeit ist immer auch eine charakteristisch geformte „Hits-Kurve“ enthalten (s. Abb. 17 auf Seite 73), die den Anteil der Hit-Frames für die unterschiedlichen  $\gamma_r$ -Werte anzeigt. Diese Kurve lässt sich schon im Vorfeld der eigentlichen (zeitaufwändigen) Analyse schnell ermitteln (siehe 4.2: „Wolken-Plot“), und kann daher Hilfestellung bei der Wahl eines sinnvollen Wertes von  $\gamma_r$  geben:

Analog zu der Erkenntnis über den Wert des optimalen  $d$  Parameters in [5] kann auch für die hier verwendete Radien-Schaar mittels der Hits-Kurve ein sinnvoller  $\gamma_r$  Parameter abgeleitet werden. Von Boulougouris wurde der Durchmesser  $d$  der Hartkugel „etwa

10% größer als die dichteste Annäherung zwischen Matrix und Penetrant“ gewählt. Diese dichteste Annäherung ist auch aus der Hits-Kurve ablesbar: Es ist der Wert von  $\gamma_r$  bei dem 100% Hits erreicht werden; er wird mit  $\gamma_r^{100\%}$  bezeichnet. Die zehnpromzentige Vergrößerung des  $d$ -Wertes teilt sich nach Gleichung 25 in zwei Anteile auf: eine Multiplikation mit 1.1 für  $\gamma_r^M$  (wodurch der Penetrant-Radius direkt um 10% vergrößert wird) und einen absoluten Betrag von +0.1 (der die Vergrößerung des jeweiligen Matrix-Radius bewirkt). Der dieser Art bestimmte Penetrant-Radien Skalierungsfaktor  $\gamma_r^M$ , bei dem die Messung (Auswertung der Simulation) erfolgt, beträgt somit:

$$\gamma_r^M \approx 1.1 \cdot \gamma_r^{100\%} + 0.1 \quad (26)$$

(Hierbei wird angenommen, dass Penetrant- und Matrix-Radien nicht übermässig voneinander abweichen.)

Die nach Gleichung (26) ermittelten Penetrant-Radien Skalierungsfaktoren führten in den Modellen dieser Arbeit zu Hit-Raten von 65 bis 85%, also in dem weiter oben als sinnvoll beurteilten Bereich.

### 3.1.6 Ermittlung von repräsentativen Hartkugelradien für einzelne Atome

Für die geometrische Bestimmung des zugänglichen Volumens  $\phi$  und des Penetrant-Matrix Überlappungs-Kriteriums  $H$  müssen die Atome des Gesamtsystems formal in (zum Teil sich durchdringende) harte Kugeln umgewandelt werden. Jedem Atom wird daher ein individueller Radius  $r_n$  zugewiesen, der vom van-der-Waals Kraftfeld-Parameter (van-der-Waals Radius) seines Atomtyps im verwendeten Kraftfeld abhängt. Der Radius ist so gewählt, dass sich die nichtbindende Wechselwirkungsenergie für ein Paar gleicher Atome mit Typ  $n$  zu Null ergibt:  $E(2r_n) = 0$ .

Im Falle eines (12,6)-Lennard-Jones Potenzials, wie es beispielsweise im CVFF-Kraftfeld in der Form  $E(r) = (A/r^{12}) - (B/r^6)$  Verwendung finden, bestimmt sich der Hartkugelradius wie folgt:

$$r_n = \frac{1}{2} \sqrt[6]{\frac{A}{B}} \quad (27)$$

Für ein (9,6)-Potenzial (verwendet beispielsweise in PCFF und COMPASS<sup>38</sup>) mit  $E(r) = \varepsilon[2(\sigma/r)^9 - 3(\sigma/r)^6]$  gilt:

$$r_n = \frac{1}{2} \sigma / \sqrt[3]{\frac{3}{2}} \quad (28)$$

Eine Dritte Methode, den Hartkugel-Radius für die Atome zu ermitteln besteht darin, die entsprechenden  $r_n$ -Werte aus einer externen Tabelle abzulesen. (Diese Methode wird vornehmlich dann verwendet, wenn detaillierte Zwischenergebnisse mit früheren Berechnungen verglichen werden sollen, bei denen nur elementspezifische Radien verwendet wurden.)

### 3.1.7 Anbindung der DPD Berechnungsmethode an *Discover*

Hierzu ist zunächst eine Übergabe der geometrischen Konfiguration (Koordinaten und Atomtypen) aller Frames des untersuchten Systems notwendig. Die Informationen werden durch ein in BTCL<sup>39</sup> geschriebenes, sog. „Überführungs-Script“ aus den ACCEL-RYS-eigenen Trajektorien-Dateien gelesen und in ein leichter bearbeitbares Klartext-Format aufbereitet. Dabei werden gleichzeitig schon die van-der-Waals Parameter der jeweiligen Atomtypen aus dem verwendeten Kraftfeld in Hartkugelradien umgerechnet. Aufwändiger ist danach die Ermittlung der Solute-Energie, da diese nicht direkt aus der Trajektorie gelesen werden kann:

**Ermittlung der Solute-Energie** Der Beitrag  $U_2^{(1)}$  in Gleichung (23) beinhaltet (wie in der Widom Methode, vgl. Seite 26) die potenzielle Energie des Penetrant Moleküls aufgrund seiner Wechselwirkung mit der Matrix, die dem System verloren geht wenn der Penetrant entfernt wird (siehe auch Gleichungen (18) und (19)). Da zwischen Penetrant und Matrix keine chemischen Bindungen existieren, wird die Wechselwirkung

<sup>38</sup>vgl. Gleichung (68m) in Anhang A.3.

<sup>39</sup>BTCL ist die scripting-Sprache von *Discover*. Es handelt sich dabei um eine Variante der verbreiteten TCL-Programmiersprache [65], die für den Umgang mit den Daten einer atomistischen Simulation erweitert wurde.



ausschließlich durch die nicht-bindenden Wechselwirkungsenergien (Coulomb und van-der-Waals) des Systems beschrieben.

Die *Discover* Simulations-Software berechnet während einer MD-Simulation routinemäßig alle Anteile der potenziellen Energie in einer Systemkonfiguration und speichert die Einzelbeiträge (hierarchisch angeordnet) für jedes einzelne Atom in der Energie-Datenbank. Zur Ermittlung von  $U$  müssen daher die passenden Beiträge der nichtbindenden Energie-Anteile für alle Atome des Penetrant-Moleküls ausgelesen und summiert werden. Diese Aufgabe übernimmt ebenfalls das o. g. Überführungs-Script. Der zur Energie-Berechnung benötigte Teil des relativ umfangreichen Scripts ist in Anhang A.5.2 gezeigt und erläutert.

## 3.2 Diskussion der statistischen Genauigkeit der Methode

Im Gegensatz zur theoriekonformen Berechnung des chemischen Potenzial gemäß Gleichung (23) aus dem „exakten“ Erwartungswert  $\langle A \rangle$  einer dynamischen Variable  $A$  wird in der praktischen Durchführung (siehe Seite 26) nur der *zeitliche* Mittelwert  $\bar{A}$  einer begrenzten Anzahl von  $A(t)$ -Werten verwendet. Daher ist der berechnete Wert für  $\mu^{\text{ex}}$  mit einer statistischen Unsicherheit (einem Fehler) behaftet der von der Differenz zwischen Erwartungswert und Mittelwert  $|\langle A \rangle - \bar{A}| = \Delta\bar{A}$  herrührt:

$$\mu^{\text{ex}} = \mu^{\text{ex}}(\langle A \rangle) \quad (29)$$

$$= \mu^{\text{ex}}(\bar{A} \pm \Delta\bar{A}) \quad (30)$$

$$[ = \mu^{\text{ex}}(\bar{A}) \pm \Delta\mu^{\text{ex}}(A(t)) ] \quad (31)$$

### 3.2.1 Standardabweichung

In der Statistik wird solch ein Fehler  $\Delta\bar{A}$  durch die *Standard-Abweichung*  $\sigma$  beschrieben, die sich als Wurzel aus der *Varianz* (dem mittleren quadratischen Fehler)  $\sigma^2$  berechnet:

$$\Delta\bar{A} \equiv \sigma_A = \sqrt{\sigma_A^2} \quad ; \quad \sigma_A^2 \equiv \overline{(A - \bar{A})^2} \quad \text{mit} \quad \bar{A} = \frac{1}{n} \sum_{t=1}^n A(t) \quad (32)$$

(Hierbei wird vorausgesetzt, dass die Einzelwerte symmetrisch um den Mittelwert verteilt sind, und idealerweise eine *Normalverteilung* (Gauß'sche Glockenkurve) darstellen.)

Der Mittelwert  $\bar{A}$  und seine Standardabweichung  $\sigma_A$  sind spezifisch für die konkret untersuchte Stichprobe an  $A(t)$ -Werten (Trajektorie). Für eine zweite Simulation wird also in der Regel ein leicht unterschiedlicher Mittelwert berechnet werden, von dem man bei einer längeren Simulation auch erwartet, dass er genauer sein wird. Die Standardabweichung (Breite der Normalverteilung) hingegen ist qualitativ unabhängig von der Menge der  $A(t)$ -Werte ist: eine längere Simulation verkleinert diese Kenngröße für die Streuung also nicht (vgl. 3.2.4).

### 3.2.2 Standardabweichung bei Löslichkeitskoeffizienten

Zur unmittelbaren Berechnung des Löslichkeitskoeffizienten müssen die Gleichungen (16) und (23) zusammengefasst werden. Es ergibt sich:

$$S = C \cdot \exp(-\beta\mu) \quad (33)$$

$$= C \cdot \exp(-\ln(\langle B \rangle \langle A \rangle^{-1})) \quad (34)$$

$$= C \cdot \langle A \rangle / \langle B \rangle \quad (35)$$

Es gehen also zwei Erwartungswerte direkt in die Berechnung der Löslichkeit ein. Diese Werte sind fehlerbehaftet, weil praktisch nur ein Mittelwert über eine begrenzte Anzahl von Konfigurationen verwendet werden kann. Konkret handelt es sich um die folgenden beiden Mittelwerte:

- $\langle A \rangle \equiv \langle \frac{1}{V} \rangle = (\bar{A} \pm \sigma_A)$

Wobei das Volumen eines equilibrierten Packungsmodells bei einer typischen NpT MD-Simulation nur sehr wenig schwankt (und bei einer NVT MD-Simulation notwendigerweise konstant ist). Die Standardabweichung von  $A$  kann daher problemlos nach Gleichung (32) bestimmt werden.

Für die *online*-Berechnung der aktuellen Standardabweichung während des Kumulierens der Einzelwerte für die DPD Mittelwertbildung wird der alternative Ausdruck für die Varianz  $\sigma^2$  verwendet:

$$\sigma_A^2 = \overline{(A - \bar{A})^2} = \overline{A^2} - \bar{A}^2 = \frac{\sum(A^2)}{n} - \left(\frac{\sum A}{n}\right)^2 \quad (36)$$

der wiederum ohne Kenntnis des endgültigen Mittelwertes  $\bar{A}$  auskommt, da die beiden enthaltenen Mittelwerte über  $A^2$  bzw.  $A$  unabhängig voneinander kumuliert werden. (Laut Definition von  $\sigma^2$  in Gleichung (32) kann der äußere Mittelwert eigentlich erst dann bestimmt werden, wenn  $\bar{A}$  nach einer ersten Bearbeitung aller  $A(t)$ -Werte bekannt ist.)

- $\langle B \rangle \equiv \left\langle \frac{\Pi H \exp \beta U}{\phi} \right\rangle = (\bar{B} \pm \sigma_B)$

Hierbei ist die Besonderheit zu berücksichtigen, dass es zwei formal unterschiedliche Typen von  $B$ -Beiträgen zum Mittelwert gibt die sich durch den Wert für  $\Pi H$  (*Null* oder *Eins*) unterscheiden:

$$B'(i = 1, \dots, h) = B(j = 1, \dots, n) \text{ mit } \Pi H = 1 \quad (37)$$

$$B''(i = 1, \dots, z) = B(j = 1, \dots, n) \text{ mit } \Pi H = 0 \quad (38)$$

Für eine Trajektorie mit  $n$  Konfigurationen gibt  $z$  die Anzahl der Konfigurationen an, bei denen  $\Pi H=0$  („zero“) ist. Hierbei handelt es sich um exakt bekannte Beiträge mit einem Wert von  $B''=0$ , die *keine* Streuung aufweisen: also auch  $\sigma_{B''}=0$ .

Die Beiträge der restlichen  $h$  Konfigurationen (die so genannten „hit-Frames“ mit  $\Pi H=1$ ) weisen eine statistische Streuung auf, deren Varianz  $\sigma_{B'}^2$  sich analog zu Gleichung (36) bestimmen lässt:

$$\sigma_{B'}^2 = \overline{B'^2} - \bar{B}'^2 = \frac{\sum (B'^2)}{n} - \left( \frac{\sum B'}{n} \right)^2 \quad (39)$$

Die Kombination der  $h+z$  Beiträge  $B'$  und  $B''$  zum Mittelwert  $\bar{B}$  stellt sich nun wie folgt dar:

$$\bar{B} = \frac{1}{n} \sum_n B = \frac{1}{h+z} \left( \sum_h B' + \sum_z B'' \right) \quad (40)$$

$$= \frac{1}{h+z} \left( \sum_h B' + 0 \right) \quad / \cdot \frac{h}{h} \quad (41)$$

$$= \frac{h}{h+z} \cdot \frac{1}{h} \sum_h B' \quad (42)$$

$$= \frac{h}{h+z} \cdot \bar{B}' = \frac{h}{n} \cdot \bar{B}' \quad (43)$$

Die Hit-Rate  $\frac{h}{n}$  stellt also den Faktor dar, der den Mittelwert  $\overline{B'}$  der *Hit-Frame* Beiträge mit dem Mittelwert  $\overline{B}$  über alle Trajektorien-Frames verknüpft. Im Rahmen der Fehlerrechnung wird dieser Faktor auch auf den Fehler von  $\overline{B'}$  angewendet. Mit anderen Worten: der *relative* Fehler von  $\overline{B'}$  bleibt unverändert, wenn mit der Konstanten multipliziert wird um  $\overline{B}$  zu erhalten:

$$\sigma_B = \frac{h}{n} \cdot \sigma_{B'} \quad (44)$$

In den hier durchgeführten NpT Simulationen ist der durch Volumen-Fluktuation bedingte Fehler gut zwei Größenordnungen kleiner als der Volumen-Beitrag selbst:  $\sigma_A \approx 7.5 \cdot 10^{-2} \overline{A}$ . Die beobachteten Werte von  $\sigma_B$  waren hingegen häufig größer als  $\overline{B}$  selbst; mit anderen Worten: Der Fehler war größer als 100% und ragte somit bis in den negativen Bereich von  $B$  hinein; was physikalisch unsinnig ist, denn dies würde in letzter Konsequenz einem „negativen Löslichkeitskoeffizienten“ entsprechen. Ursache hierfür ist die unsymmetrische Verteilung<sup>40</sup> der  $B$ -Beiträge, wie in Abbildung 6 exemplarisch gezeigt.

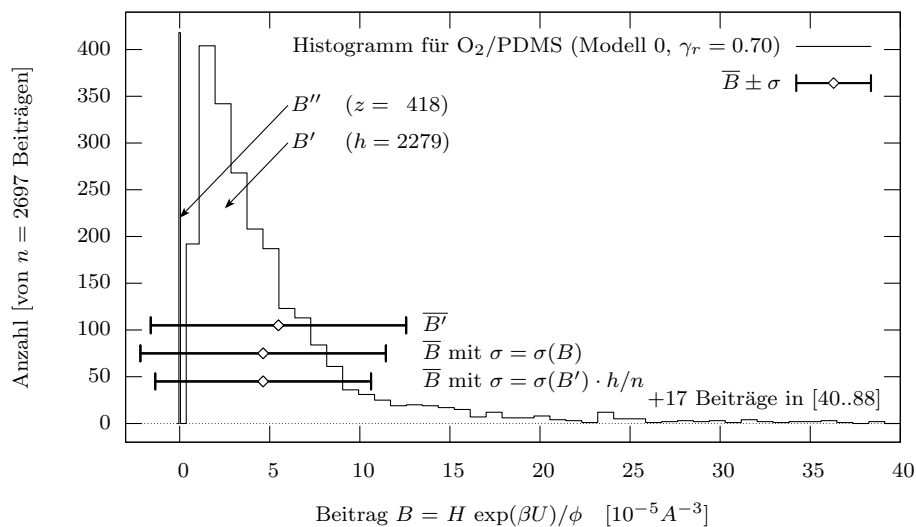
⇒ Für  $B$  und den Löslichkeitskoeffizienten nach Gleichung (35) kann kein sinnvoller, *symmetrischer* Fehler-Bereich berechnet werden. Höchstens ein „Streu-Bereich“  $S^\pm$  kann aus den Einzel-Löslichkeiten abgelesen werden, um eine obere Grenze  $S^+$  und eine untere Grenze  $S^-$  der berechneten Löslichkeiten zu beschreiben.

### 3.2.3 Standardabweichung des chemischen Exzesspotenzials

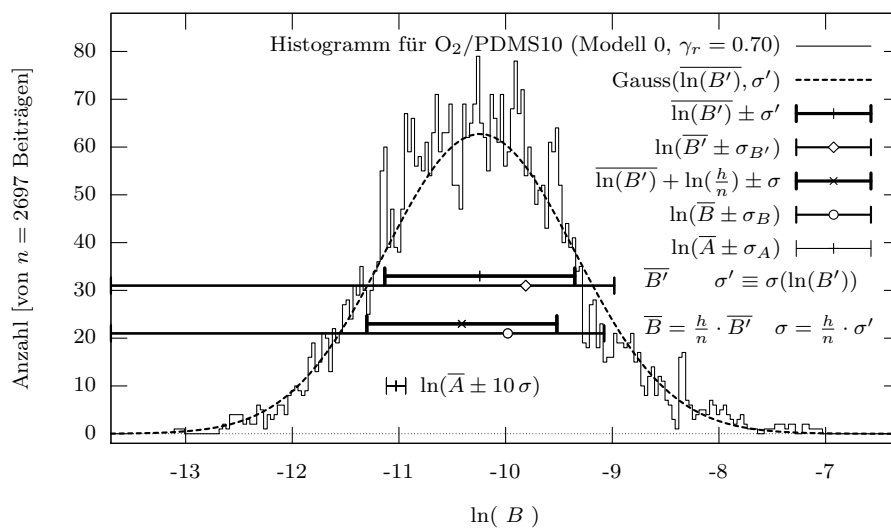
Für das chemische Potenzial hingegen kann ein symmetrischer Fehler mit Hilfe der Standardabweichung angegeben werden, da hier nicht  $\langle B \rangle$  selbst sondern  $\ln\langle B \rangle$  verwendet wird (vgl. Gleichung (34)):

$$\beta\mu = \ln(\langle B \rangle \langle A \rangle^{-1}) = \ln\langle B \rangle - \ln\langle A \rangle \quad (45)$$

$$= [\ln(\overline{B}) - \ln(\overline{A})] \pm [\sigma(\ln(B)) + \sigma(\ln(A))] \quad (46)$$



**Abbildung 6:** Das Histogramm der Einzelbeiträge zu  $\bar{B}$  zeigt eine unsymmetrische Verteilung.



**Abbildung 7:** Das Histogramm der Logarithmen der Einzelbeiträge zu  $\bar{B}'$  ( $\sim \beta\mu$ ) zeigt eine symmetrische (Normal-) Verteilung.

Wie Abbildung 7 zeigt, sind die Logarithmen der nicht-Null Beiträge  $B'$  in guter Näherung normalverteilt, so dass für diese eine sinnvolle Standardabweichung  $\sigma(\ln(B'))$  (kurz:  $\sigma'$ ) berechnet werden kann:

$$\sigma' = \sigma(\ln(B')) = \sqrt{\frac{\sum(\ln(B')^2)}{n} - \left(\frac{\sum \ln(B')}{n}\right)^2} \quad (47)$$

Zur Übertragung des Fehlers von  $\ln(B')$  auf  $\ln(B)$  muss die in Gleichung (43) genannte Beziehung zwischen  $\bar{B}$  und  $\bar{B}'$  verwendet und ebenfalls logarithmiert werden, da die direkte Berechnung aus den einzelnen  $\ln(B(t))$  Beiträgen nicht möglich<sup>41</sup> ist.

$$\sigma(\ln(\bar{B})) = \sigma(\ln(\bar{B}' \cdot \frac{h}{n})) \quad (48)$$

$$= \sigma(\ln(\bar{B}') + \ln(\frac{h}{n})) \quad (49)$$

$$= \sigma(\ln(\bar{B}')) + \sigma(\ln(\frac{h}{n})) \quad (50)$$

$$= \sigma(\ln(\bar{B}')) + 0 = \sigma' \quad (51)$$

Der zweite Summand in Gleichung (50) stellt die Streuung einer festen Zahl dar und ist somit Null. Bildhaft gesprochen verschiebt die Addition von  $\ln(\frac{h}{n})$  in Gleichung (49) das Histogramm nur seitwärts, hat aber keinen Einfluss auf die Form (Breite) des durch  $\sigma'$  beschriebenen  $\ln(B)$ -Histogramms.

Bemerkung 1: Im  $\ln(B)$ -Histogramm (Abb. 7) ist zu beobachten, dass die Logarithmen der Mittelwerte sich systematisch von Mittelwerte der Logarithmen unterscheiden: ( $\ln(\bar{B}) \neq \overline{\ln(B)}$ ). Dies erklärt sich daraus, dass sich das Gewicht der numerisch kleineren  $B$ -Beiträge nicht-proportional vergrößert, wenn die  $B$ -Beiträge zuerst logarithmiert werden. Daher werden die Einzelbeiträge zu  $B$  zuerst gemittelt und dann logarithmiert.

Bemerkung 2: Die typische Standardabweichung von  $\ln(A)$  ist erheblich kleiner als  $\sigma'$ . Daher werden  $\sigma_A$  bzw.  $\sigma_{\ln(A)}$  in den weiteren Berechnungen vernachlässigt.

⇒ Für das hier berechnete chemische Exzesspotenzial werden die fehlerbehafteten Erwartungswerte in Form der Mittelwert  $\ln(\bar{B})$  und  $-\ln(\bar{A})$  zusammen mit der Standardabweichung  $\sigma' = \sigma_{\ln(B')}$  verwendet.

<sup>40</sup>Es handelt sich hier um eine sog. „Log-Normal“ Verteilung.

<sup>41</sup>Für die  $B''$ -Beiträge müsste  $\ln(0)$  berechnet werden, welches undefiniert ist.

### 3.2.4 (Minimaler) Standardfehler

Um die Genauigkeit der Berechnungen in Form eines Fehlerwertes  $\Delta\bar{A}$  zu beschreiben, ist im Vergleich zur oben verwendeten Standardabweichung der statistische Standardfehler  $\langle\sigma_A\rangle$  (oder  $\hat{\sigma}_A$ ) grundsätzlich besser geeignet. Er beschreibt den Erwartungswert für die Abweichung zwischen mehreren unabhängigen Mittelwerten einer dynamischen Variable  $A(t)$ . Die Herleitung von  $\hat{\sigma}_A$  als Wurzel aus der mittleren quadratischen Abweichung der Mittelwerte vom Erwartungswert in [66] ergibt:

$$\hat{\sigma}_A = \langle\sigma_A\rangle = \langle\Delta\bar{A}\rangle = \sqrt{\langle(\bar{A} - \langle A \rangle)^2\rangle} = \sqrt{\frac{2}{N_\tau}} \cdot \sqrt{\langle(A - \langle A \rangle)^2\rangle} \quad (52)$$

$$\approx \sqrt{\frac{2}{N_\tau}} \cdot \sqrt{(A - \bar{A})^2} \quad (53)$$

$$= \sqrt{\frac{2}{N_\tau}} \cdot \sigma_A \quad (54)$$

Wobei  $N_\tau (=t_{\text{sim}}/\tau_A)$  die Anzahl der Korrelations-Zeiten  $\tau_A$  beschreibt, die die vorliegende Simulation  $A(t)$  beinhaltet. Mit anderen Worten: Wie oft während einer Simulation der Länge  $t_{\text{sim}}$  die dynamische Variable  $A$  üblicherweise um ihren Erwartungswert  $\langle A \rangle$  schwankt. Der Vorteil von Gleichung (53) besteht nun darin, dass keine Kenntnis des wahren Erwartungswertes  $\langle A \rangle$  notwendig ist, um den Standardfehler von  $\bar{A}$  zu berechnen. Allerdings muss jeweils noch  $N_\tau$  aus der Korrelations-Zeit  $\tau_A$  und der Länge  $t_{\text{sim}}$  der Simulation bestimmt werden.

In [67] wurde diesbezüglich gezeigt, dass die Streuung  $\bar{\sigma}$  aus berechneten Blockmittelwerten unabhängig von der Blockgröße ist, wenn  $t_{\text{sim}}$  des einzelnen Blocks größer als eine Korrelationszeit  $\tau_A$  ist. Um also  $\tau_A$  zu bestimmen, müsste die Datenserie  $A(t)$  in immer kleinere Blöcke zerlegt und ausgewertet werden. Diese Untersuchung wurde hier jedoch nicht durchgeführt.

Stattdessen soll ein maximaler Wert für  $N_\tau$  abgeschätzt werden, der zu einem minimalen Standardfehler führt: Bei Untersuchung einer Trajektorie mit  $h$  Hit-Frames kann  $N_\tau$  niemals größer als  $h$  sein. Konkret kann es maximal *halb* so groß wie  $h$  sein, da eine

größere Anzahl an Schwingungsperioden von  $A$  um den Mittelwert  $\bar{A}$  sonst nicht mehr erfasst werden könnte (Shannon-Abtast-Theorem). Alle in den nachfolgend genannten Standardfehler  $\hat{\sigma}$  bzw.  $\hat{S}^\pm$  sind immer solche *minimalen* Standardfehler mit  $N_\tau = h/2$ :

$$\hat{\sigma}_\mu = \sqrt{4/h} \cdot \sigma_\mu \quad (55)$$

### 3.2.5 Ausreißer

Als Ausreißer bezeichnet man einzelne Werte einer Datenmenge, die ungewöhnlich weit von allen anderen beobachteten Werten abweichen. Als Kriterium, was als „weit“ zu verstehen ist, wird bei einer Normalverteilung die Standardabweichung  $\sigma$  herangezogen: Bei einem Datenwert, der weiter als die  $4\sigma$  vom Mittelwert abweicht, spricht man von einem *milden* Ausreißer (Wahrscheinlichkeit 1:150). Wenn die Abweichung mehr als  $7\sigma$  beträgt, liegt ein *extemer* Ausreißer vor (Wahrscheinlichkeit 1:425 000).

Da die genannten Grenzen für Ausreißer nur für eine Normalverteilung gültig sind, wird der Ausreißertest in dieser Arbeit für die  $\ln(B')$ -Werte durchgeführt, aus denen auch die Standardabweichung der Gesamtberechnung abgeleitet wurde. Hierbei fanden sich in den untersuchten Modellen (jeweils 2800 Frames) zwischen Null und Drei milde Ausreißer und keine extremen Ausreißer. Daher wurde die Ausreißerproblematik nicht weiter verfolgt, denn diese Anzahl an milden Ausreißern darf für 2800 Frames erwartet werden. (Es wird vermutet, dass sich durch die Entfernung der milden Ausreißer die starken Stufen in den Radian-Plots<sup>42</sup> verringert würden; allerdings sollten Ausreißer grundsätzlich erst nach genauer Ursachenermittlung entfernt werden.)

## 3.3 Analytische Berechnung des Volumens in atomistischen Polymer–Penetrant Packungsmodellen

### 3.3.1 Atomistische Volumen-Nomenklatur

Das *freie* Volumen  $V_F$  eines Packungs-Modells erhält man, indem vom geometrischen Volumen der gesamten kubischen Packung ( $V = a \cdot b \cdot c$ ) das durch alle Matrix-Atome

---

<sup>42</sup>Beispielsweise nach unten bei  $\gamma_r=0.65$  in Abb. 34 (Seite 103) oder nach oben in Abb. 29 (Seite 96) bei  $\gamma_r=0.54, 0.64$  und  $0.73$



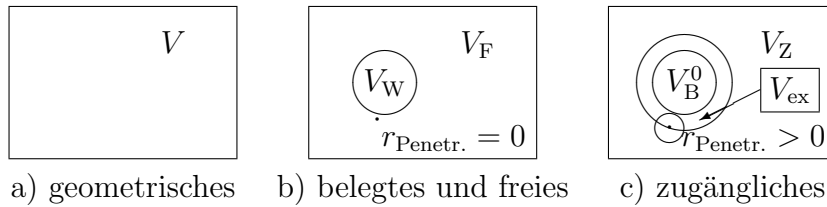
der Packung bereits belegte van-der-Waals Volumen  $V_W$  abgezogen wird (Gleichung (1), siehe Abb. 8a, b).

Dieses freie Volumen steht jedoch nur einem imaginären, *punktförmigen* Teilchen vollständig zur Verfügung. Für die in dieser Arbeit durchgeführten Berechnungen mittels der DPD Methode wird hingegen das *zugängliche* Volumen  $V_Z$  benötigt, welches den Radius des eingefügten Probe-Teilchens (Penetrant) berücksichtigt (siehe Abb. 8c). Praktisch wird im Fall eines einatomigen Penetranten zur Berechnung des zugänglichen Volumens der Radius jeder Matrix-Kugeln um den Radius des Penetranten vergrößert [6], so dass sich das *belegte* (van-der-Waals) Volumen der Matrix  $V_W = V_B^0$  um ein *zusätzliches* Volumen  $V_{\text{ex(zess)}}$  zum *blockierten* Volumen  $V_B$  vergrößert:

$$V_Z = V - [V_B^0(r_{\text{Matrix}}) + V_{\text{ex}}] \quad (56)$$

$$= V - V_B(r_{\text{Matrix}} + r_{\text{Penetrant}}) \quad (57)$$

$V_Z$  umfasst somit all diejenigen Punkte im Raum, an denen sich das Zentrum des Penetranten befinden kann, ohne dass die Penetrant-Kugel mit einer Matrix-Kugel überlappt.



**Abbildung 8:** Übersicht zur Volumen-Nomenklatur

Aus dem zugänglichen Volumen wird schließlich noch der *Anteil*  $\phi$  an zugänglichem Volumen in der Packung berechnet:

$$\phi \equiv \frac{V_Z}{V} = 1 - \frac{V_B(\text{Penetrant})}{V} \quad (58)$$

Handelt es sich beim Penetranten nicht um eine einfache Kugel, so reicht die Addition von Matrix- und Penetrant-Radien gemäß (57) nicht aus, um  $V_B$  zu bestimmen. Vielmehr wird das zusätzliche Volumen  $V_{\text{ex}}$  nun auch noch von der Struktur des mehratomigen Penetranten bestimmt:

### 3.3.2 Zusätzliches Volumen bei mehratomigen Penetranten

Ein dreiatomiger Penetrant „ $A-B-C$ “ wird durch seine Radien ( $r_A, r_B, r_C$ ) und die Verbindungsvektoren ( $\overrightarrow{AB}, \overrightarrow{BC}$ ) seiner Atome beschrieben. (Alternativ können auch die Ortsvektoren  $\vec{r}_A, \vec{r}_B$  und  $\vec{r}_C$  oder die Verbindungsvektoren  $\overrightarrow{SA}, \overrightarrow{SB}, \overrightarrow{SC}$  der Atome mit einem beliebigen Punkt „ $S$ “ (Ursprung, Schwerpunkt, erstes Penetrant-Atom, o.ä.) verwendet werden um die Struktur zu beschreiben.)

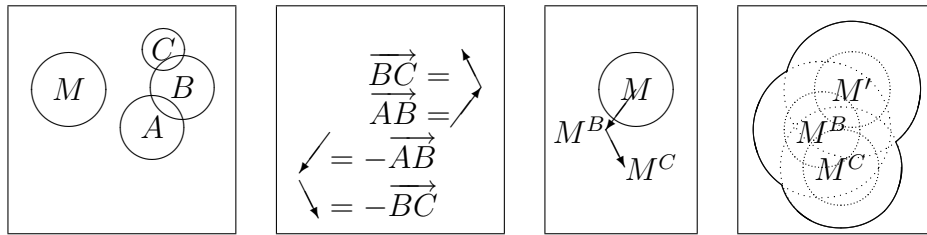
Zur Ermittlung des blockierten Volumens  $V_B(M, ABC)$ , für den Fall einer Matrix, die aus einer einzigen Kugel „ $M$ “ besteht, werden zunächst *Kopien* ( $M^B, M^C$ ) der ursprünglichen Matrix-Kugel  $M$  angelegt, deren Position durch die *inversen* Verbindungsvektoren des Penetranten bestimmt sind [6]:

$$\overrightarrow{MM^B} = -\overrightarrow{AB} \quad ; \quad \overrightarrow{M^B M^C} = -\overrightarrow{BC}$$

Anschließend wird der Radius der ursprünglichen Matrix-Kugel  $M$  um den Radius der Basis-Kugel  $A$  des Penetranten vergrößert. Die Matrix-Kopien ( $M^B, M^C$ ) werden um den Radius der jeweils zugrundeliegenden Penetrant-Kugel ( $B$  bzw.  $C$ ) vergrößert.

$$r_{M'} = r_M + r_A \quad ; \quad r_{M^B} = r_M + r_B \quad ; \quad r_{M^C} = r_M + r_C$$

Der gesamte Vorgang ist in Abbildung 9 skizziert.



Matrix und Solute    Solute Struktur    Kopie-Positionen und -Radien

**Abbildung 9:** Blockiertes Volumen bei mehratomigen Penetranten

Das blockierte Volumen für einen mehratomigen Penetranten besteht somit aus einer Anordnung mehrerer sich durchdringender Kugeln im Raum. Die rechnerische Ermittlung des Volumens einer solchen *Schnittmenge* von Kugeln  $V(\cup \text{Kugeln})$  gemäß Gleichung (59) ist in Abschnitt 3.3.3 beschrieben.

Die praktische Ermittlung der *zusätzlichen* Positionen der Kugeln im Programm erfolgt nicht durch die der Bindungs-Vektoren des Penetrant-Moleküls (wie in Abbildung 9 gezeigt), sondern es werden die (inversen) Verbindungs-Vektoren aller Penetrant-Kugeln zum gemeinsamen geometrischen Mittelpunkt des Penetranten ( $\vec{r}_{j=0} \equiv \sum_{j=1}^{N_2} \vec{r}_j / N_2$ ) für die Verschiebung verwendet. Der praktische Nutzen hiervon ist, dass es kein ausgezeichnetes „Start-Atom“ des Penetranten gibt, und auch Ring-Strukturen im Penetrant vorhanden sein dürfen, ohne eine Sonderbehandlung zu benötigen. Der vollständige Ausdruck des durch die Matrixatome  $i$  für einen  $j$ -atomigen Penetranten blockierten Volumens  $V_B$

$$V_B = V \left( \bigcup_{\substack{i=1, \dots, N_1 \\ j=1, \dots, N_2}} \text{Kugel} \left[ \begin{array}{l} \vec{r}_i - (\vec{r}_j - \vec{r}_{j=0}) \\ r_i + \gamma_r r_j \end{array} \right] \right) \quad (59)$$

beinhaltet neben der Berücksichtigung des mehratomigen Penetranten aus  $N_2$  Atomen auch zugleich den Einfluss aller  $N_1$  Matrix-Atome und den Penetrant-Radien Skalierungsfaktor  $\gamma_r$  (vgl. Abschnitt 3.1.5).

### 3.3.3 Praktische Volumenberechnung großer Kalottenmodelle

In der Hartkugel-Darstellung durchdringen sich die Atome eines Moleküls oder Polymers teilweise, so dass jedes Atom üblicherweise weniger Raum als sein rechnerisches Kugelvolumen belegt. Die Situation ist dabei absolut vergleichbar mit dem rechten Teilbild von Abbildung 9, was es im Umkehrschluß ermöglicht, mit den gleichen Methoden das belegte Volumen einer Polymerkette für einen sphärischen oder für einen mehratomigen Penetranten zu berechnen; im letzteren Fall gehen nur mehr Kugeln in die Berechnung ein.

Eine grundlegende Methode zur effizienten Berechnung des belegten Volumens einer beliebigen Mischung von sich (teilweise) durchdringenden Kugeln wurde 1991 von Dodd und Theodorou [60] beschrieben. Im Folgenden werden die Grundzüge dieser Methode erläutert, mit der sich das Volumen von Kalottenmodellen berechnen läßt.

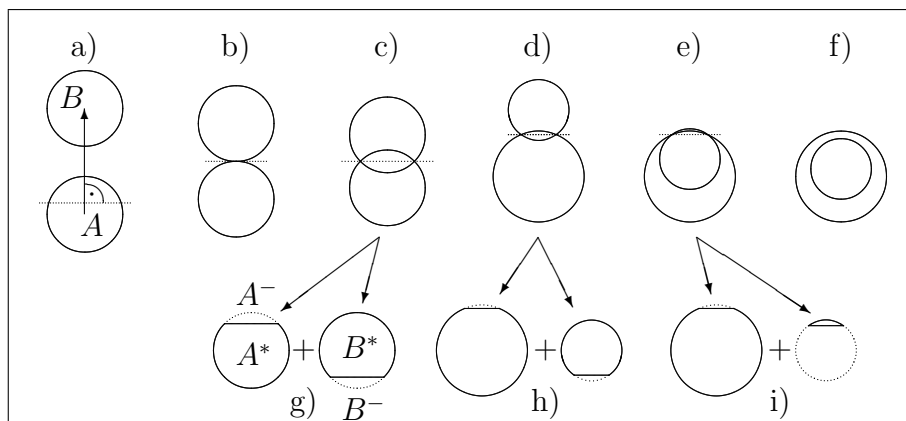
**Methode von Dodd und Theodorou** Das belegte Gesamtvolumen eines Systems sich überlappender Kugeln setzt sich zusammen aus den Anteilen des Volumens, das jede einzelne Kugel belegt. Dieses Volumen  $V_{\Delta}(i)$  das jede einzelne Kugel  $i$  belegt, ist üblicherweise geringer als deren geometrisches Volumen ( $V_{\text{Kugel}} = \frac{4}{3}\pi r^3$ ), da Teile der Kugel mit anderen Kugeln überlappen. Die Überlappungen werden durch Schnittebenen beschrieben:

Zwischen zwei beliebigen Kugeln beschreibt der Verbindungsvektor  $\overrightarrow{AB}$  (s. Abb. 10a) der Kugelzentren die Orientierung einer Ebenenschar senkrecht zu diesem Vektor; durch Auswahl eines Punktes auf der Verbindungslinie wird genau eine dieser Ebenen festgelegt. Bei sich durchdringenden Kugeln läßt sich dieser Punkt (in Abhängigkeit von den Kugel-Radien und dem Abstand der Kugelzentren) genau so wählen, dass die durch ihn selektierte Ebene den Schnittkreis der Kugeloberflächen beinhaltet (Abb. 10c–e). Diese Ebene trennt die beiden Kugeln ( $A$  und  $B$ ) dann in jeweils zwei Teile: Einen der anderen Kugel abgewandten Anteil (Abb. 10g:  $A^*$  und  $B^*$ ) und einen der anderen Kugel zugewandten Anteil ( $A^-$  und  $B^-$ ). Solch eine Ebene existiert bei fast allen denkbaren Arten der Durchdringung (vom knappen Hineinragen: Abb. 10d, bis zum leichten Hinausragen: Abb. 10e), nicht jedoch bei Kugeln, die identische Position und Radien haben oder bei denen eine Kugel komplett innerhalb der anderen Kugel liegt (Abb. 10f); Kugeln, die sich nicht berühren (Abb. 10a), haben selbstverständlich auch keine solche Ebene.

Bei der Betrachtung von Atom-Packungen unter periodischen Randbedingungen (siehe Abschnitt 2.4.2) existieren weitere Ebenen (im kubischen Falle sechs), die ebenfalls einzelne Kugel aus dem Inneren der Packung schneiden und in zwei Anteile (Innen und Außen) trennen.

Die Ebene trennt jeweils „Kappen“ ( $A^-$ ,  $B^-$ ) von den Kugeln ab, deren belegtes Volumen (und Oberfläche) fortan bei der Summierung aller belegten Volumina (und Oberflächen) nicht mehr berücksichtigt wird, da es bereits durch die jeweils andere Kugel belegt ist. Eine solche abgetrennte Kappe kann durchaus den größeren Teil einer Kugel

ausmachen (vgl. Abb. 10i); ausschlaggebend ist hierbei die Richtung des oben genannten Vektors  $\overrightarrow{AB}$  von der beschnittenen Kugel hin zur Nachbarkugel<sup>43</sup>: Der Vektor legt durch seine Richtung einen „Innenbereich“ und einen „Außenbereich“ bezüglich der Ebene fest<sup>44</sup>. Nur derjenige Kugelteil, der „Innen“ liegt geht in die Summation ein.



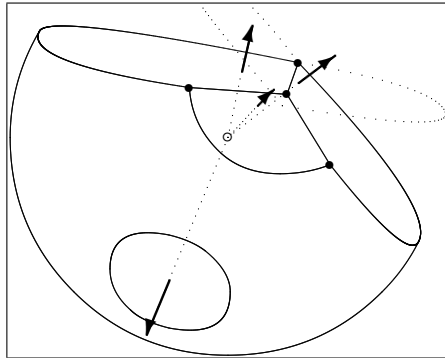
**Abbildung 10:** Relative Lage zweier Kugeln zueinander, die gebildete Durchdringungs-Ebene und die sich ergebenden, beschnittenen Kugeln

Da jede einzelne Kugel üblicherweise von *mehreren* anderen Kugeln durchdrungen wird, gibt es mehrere Ebenen, die Kappen von der Kugel abtrennen. Vielfach durchdringen sich diese Kappen dabei zumindest teilweise, so dass komplizierter geformte Fragmente von einer Kugel angetrennt werden: Die sich durchdringenden, abtrennenden Ebenen erzeugen dann Kanten (zwei Ebenen schneiden sich) oder sogar Spitzen (drei Ebenen schneiden sich) auf dem verbleibenden Fragment der Ursprungskugel (siehe Abbildung 11).

Nachdem für jede Kugeln alle abtrennenden Ebenen ermittelt wurden, kann das Volumen  $V_{\Delta}$  der verbleibenden Kugel-Reste (als Beiträge zum belegten Volumens der kompletten Packung) bestimmt werden, indem der verbliebene Rest der Kugel in weitere  $j$

<sup>43</sup>Für die Ebenen aufgrund der periodischen Randbedingungen: vom Inneren der Packung nach Außen.

<sup>44</sup>Die Unterscheidung Innen/Außen ist aus Sicht der Nachbarkugel mit  $\overrightarrow{BA} = -\overrightarrow{AB}$  genau umgekehrt.



**Abbildung 11:** Verbleibendes Fragment einer mehrfach beschnittenen Kugel (mit Normalen-Vektoren der Schnittebenen sowie Schnittkanten und Spitze zwischen drei Schnittebenen)

Fragmente  $V_{\Delta,j}$  zerlegt wird. Die  $j$  Fragmente teilen sich gemeinsam den Mittelpunkt ihrer ursprünglichen Kugel, daher kann ihr Volumen nach  $V_{\Delta,j} = A_{\Delta,j} \cdot \frac{1}{3}h$  berechnet werden. Bei den Fragmenten handelt es sich um einen der beiden folgenden Typen:

- „Kegel-Pyramiden“,

die eine der Schnittebenen als Grundfläche<sup>45</sup> und das Kugelzentrum als Spitze haben:

$$A_{\Delta} = A_E(\text{Schnittebene}) \quad , \quad h = \text{Abstand}(\text{Ebene}, \text{Kugelzentrum})$$

und

- „sphärische Fragmente“,

welche das Volumen unter der verbleibenden, gebogenen Außenfläche der Kugel beinhalten:

$$A_{\Delta} = A_K(\text{Restkugel}) \quad , \quad h = r(\text{Kugel})$$

<sup>45</sup>Die Grundfläche kann dabei von gerade Kanten (Schnittkanten) und gebogene Kanten (Schnittkreis) begrenzt sein.

Alle genannten Vorarbeiten und die Größenbestimmung der jeweiligen Flächen  $A_E$  und  $A_K$  werden vom Dodd–Theodorou Algorithmus [60] durchgeführt. Der zugehörige Software-Code zur Volumen-Berechnung wurde von Prof. Theodorou zur Verfügung gestellt. Sein korrektes Funktionieren wurde an einigen Beispielen überprüft (siehe 4.4.1), dabei traten in seltenen Fällen reproduzierbare Fehl-Berechnungen der Kalotten-Geometrie auf, die auch von Boulougouris beobachtet wurden. Diese „Bugs“ wurden, soweit wie möglich, durch entsprechende Korrekturen im Quellcode beseitigt (siehe nächster Abschnitt). Wo dies sehr aufwändig gewesen wäre, wird die Fehlersituationen zumindest erkannt, und führt zu einer gesonderten Behandlung der Geometrie-Ergebnisse.

### 3.3.4 Umgang mit Problemen bei der Volumenberechnung

**Problem der Kugelanzahl** Die Anzahl  $N_K$  der geometrisch zu untersuchenden Kugeln beträgt aufgrund der Abbildung der Penetrant-Struktur auf die Matrix ein Vielfaches der Atomanzahl im Packungsmodell:  $N_K = N_1 \cdot N_2$ . Theoretisch könnte jede dieser Kugeln mit jeder anderen überlappen und eine Schnittebene erzeugen; zur Indizierung der Ebenen-Information wird vom Program also eine Tabelle mit  $N_K^2$  Einträgen benötigt. Die drei verwendeten Fortran-Compiler limitieren die Verwendung von Array-Größen im Common-Bereich allerdings, so dass praktisch nur Systeme mit maximal (`n_Max=`) 1365 Kugeln<sup>46</sup> direkt berechnet werden können. Wird dieses Limit überschritten, reicht die Größe der vordefinierten Ebenen-Informationen-Tabelle `Plane_List`<sup>47</sup> im Volumen-Berechnungs-Algorithmus nicht mehr aus und die Geometrie-Berechnung schlägt fehl.

Um dieses Problem grundsätzlich zu beheben, wurde die Zerlegung des geometrischen Gesamtsystems in *Slice-Würfel* (siehe 3.3.5) eingeführt. Die Größe der Slice-Würfel wird mit dem Slices-Parameter  $s$  kontrolliert. Wird dieser zu gering gewählt,

---

<sup>46</sup>Diese Zahl beinhaltet die Original-Kugeln sowie alle durch periodische Randbedingungen hinzukommenden Ghost-Kugeln.

<sup>47</sup>Das Array `Integer*4 Plane_List(0:n_Max,0:n_Max,3)` fungiert als Index in sechs weitere Tabellen `Real*8 array(Max_Planes)` in der für alle *existierende* Ebenen die Raumausrichtung etc. gespeichert sind.

tritt der Fehler „zu viele Kugeln“ erneut auf und  $s$  muss erhöht werden. (Dies hilft in der Regel auch dann, wenn die Anzahl der gefundenen Ebenen im System größer als  $\text{Max\_Plane}=100 \cdot n_{\text{Max}}$  wird. Das Limit  $\text{Max\_Plane}$  ist variierbar und definiert die Größe von mehreren anderen Tabellen, die intern bei der Volumen-Analyse verwendet werden; der Faktor 100 resultiert aus Erfahrungswerten.)

**Problem sich-berührender Kugeln** Wenn sich zwei Kugeln im System exakt berühren (Abb. 10b) gibt es zwar eine Ebene zwischen den Kugeln, allerdings trennt diese keine Kappe von den Kugeln ab. Dies ist eine Situation, die vom Original-Algorithmus nicht behandelt wird. Wenn solch eine Situation auftritt, wird nun zunächst automatisch versucht, die Radien der beteiligten Kugeln leicht zu modifizieren: 10 Prozent der Kugeln im System werden (temporär) um 0.1% vergrößert, weitere 10 Prozent der Kugeln werden um 0.1% verkleinert; danach wird erneut versucht, das belegte Volumen zu berechnen. Schlägt dies immernoch fehl, werden die dritten 10% vergrößert, und die vierten 10% wiederum verkleinert, solange, bis alle Kugeln leicht in der Größe verändert wurden. Erst wenn sich dann weiterhin zwei Kugeln exakt berühren, gilt die Volumenberechnung als fehlgeschlagen.

(Als manuelle Lösung für dieses Problem könnte für solche Frames der Skalierungsfaktor  $\gamma_r$  um einen geringen Betrag verändert werden.)

**Problem „externer“ Kugeln** Eine weiterer Bug im ursprünglich Algorithmus bewirkt, dass das belegte Volumen eines Würfels (bzw. Slice-Würfels) größer berechnet wird, als der Würfel an sich überhaupt groß ist. Als Ursache hierfür wurde im vorliegenden Programm gefunden, dass manche Kugeln nicht korrekt als „außerhalb“ des Würfels erkannt wurden, und nun formal mit z. B. -1% ihres Volumens berücksichtigt werden; der Code verwendet dann jedoch +99% des Volumens der Kugel die eigentlich außerhalb liegt. Besonders wenn die untersuchten Slice-Würfel nun auch noch sehr klein sind (siehe nachfolgenden Abschnitt), sorgt schon eine einzelne solcher externen Kugeln dafür, dass mehr Volumen belegt sein soll als überhaupt zur Verfügung steht.



Diese Fehlersituation wird jetzt vom Programm erkannt und dadurch behoben, dass solch ein kleiner, negativer Volumenanteil von unter 1% einer Kugel auf Null gesetzt wird. (Dieser Vorgang wird in einer Status-Meldung dokumentiert.)

Allerdings treten übervolle Volumen teilweise auch aus anderen, nicht nachvollziehbaren Gründen auf; in diesem Falle gilt die Volumenberechnung wiederum als fehlgeschlagen.

**Berücksichtigung von Geometriefehlern in der DPD-Auswertung** Fehlgeschlagene Volumenberechnungen werden vom Volumen-Code signalisiert und sorgen dafür, dass die betroffene Konfiguration nicht in der DPD Berechnung berücksichtigt wird (siehe Seite 87 in Abschnitt 4.3.3). Mit Ausnahme der Modelle CH<sub>4</sub>/PDMS und CO<sub>2</sub>/PDMS (für  $\gamma_r < 0.6$ , siehe Seite 101) sind in den Systemen dieser Arbeit nicht mehr als zwei Geometriefehler je Trajektorie aufgetreten; so dass eine automatische Berücksichtigung, dass Geometriefehler nur selektiv in den Hit-Frames auftreten können, nicht programmiert wurde.

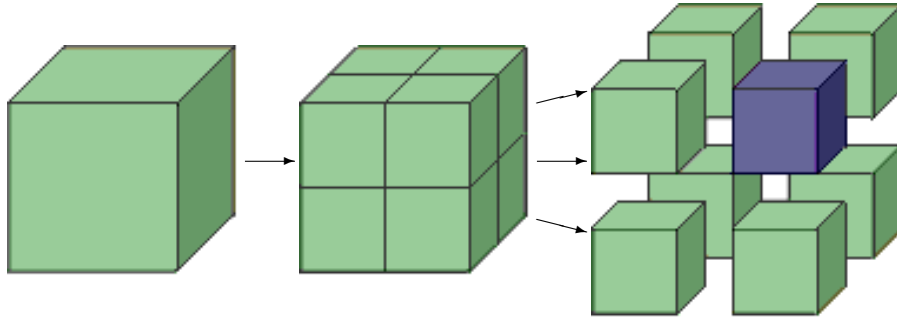
### 3.3.5 Berechnung unabhängiger Teilvolumina: *Slices*

Die Berechnungen gemäß Abschnitt 3.3.2 und 3.3.3 für eine typische Polymer-Packung mit z. B. 8500 Atomen und einem 12-Zentren Benzol als Penetrant ergeben z. B.  $102\,000 \approx 10^5$  mögliche Kugelüberlagerungen, die prinzipiell berücksichtigt werden müssen. Da sich jede Kugel mit jeder Anderen überschneiden kann, erzeugt der Algorithmus von Dodd und Theodorou zuerst eine Tabelle mit  $10^5 \cdot 10^5 = 10^{10}$  Einträgen, um alle denkbaren Überschneidungs-Paarungen zu erfassen. Dieses Array wäre nun alleine schon (je nach Datentyp) mindestens 10 GB groß, ohne dass auch nur eine einzige Volumen-Angabe enthalten wäre.

Um den Volumen-Algorithmus effizient<sup>48</sup> zu nutzen, wird daher die Analyse einer Konfiguration in kleinere Einheiten unterteilt (siehe Abb. 12). Hierzu wird die Simulationszelle zunächst geometrisch in eine gewisse Anzahl  $s$  von Scheiben (engl. *slices*) zerlegt,

---

<sup>48</sup>Speicherbedarf zur Laufzeit, Array-Limits des Compilers, Portabilität, Parallelisierbarkeit.



**Abbildung 12:** Zerlegung der Simulationszelle in 8 Slice-Würfel ( $s = 2$ )

indem (ähnlich wie bei einem Computer-Tomographen) logische Schnitte senkrecht zur x-Achse vorgenommen werden. Jede Kugel ist nun mittels ihrer x-Koordinate genau einer Scheibe zugeordnet. Das Verfahren wird in y- und z-Richtung wiederholt. Man erhält daraufhin  $s^3$  Mini-Ausgaben der Packung, die sog. „Slice-Würfel“, mit jeweils nur wenigen enthaltenen Kugeln. Zur Ermittlung des besetzten Volumens der kompletten Packung werden die Einzel-Ergebnisse aller Slice-Würfel berechnet und aufsummiert.

Die Anzahl der Slices wird nach

$$\frac{N_1 \cdot N_2}{s^3} \leq 500 \quad \Rightarrow \quad s_{\min} = \sqrt[3]{\frac{N_1 \cdot N_2}{500}} \quad (60)$$

$$s = 1 + \text{int}(s_{\min}) + s^+ \quad (61)$$

berechnet, so dass (im Falle einer homogenen Verteilung aller Kugeln) in jedem Slice-Würfel höchstens 500 Kugeln sein würden. (Zusammen mit den notwendigen benachbarten Ghost-Kugeln wird so sehr wahrscheinlich das Limit von `n.Max=1365` Kugeln nicht überschritten.) Praktisch liegt jedoch eine inhomogene Verteilung der Kugeln vor, so dass mit dem `SlicesPlus`-Parameter  $s^+$  eine größere Anzahl an Slices erzeugt werden kann.

Jeder Slice-Würfel enthält durchschnittlich  $(N_1 \cdot N_2)/s^3$  Kugeln. Wird das oben begonnene Beispiel mit einer typischen Slicezahl<sup>49</sup> von  $s=37$  fortgesetzt ergibt sich  $s^3 \approx 50\,000$

<sup>49</sup>Dieser Wert liegt nahe dem Optimum der Rechenzeit, die für die Geometrieberechnung notwendig ist. Mehr hierzu auf Seite 62.

und damit sind im Schnitt nur noch zwei (!) Kugeln in jedem Slice-Würfel zu berücksichtigen. Praktisch müssen jedoch neben den Kugeln *im* Slice-Würfel auch alle *benachbarten* Kugeln berücksichtigt werden, die sich in der Umgebung des Slice-Würfels befinden und mit ihrem Radius in den Slice-Würfel hineinragen könnten (vgl. Abschnitt 2.4.2). Dennoch ist nun die Zahl der zeitgleich zu untersuchenden Kugeln deutlich reduziert, was sich in kleineren Tabellen und auch in einer schnelleren Abarbeitung des Geometrie-Algorithmusses niederschlägt. Dieser Gewinn (siehe unten) gleicht üblicherweise den zeitlichen Mehraufwand für die Ermittlung von Inhalt und Umgebung der Slice-Würfel mehr als aus und macht größere Systeme überhaupt erst handhabbar.

### 3.3.6 Erkennen eingeschlossener Slice-Würfel: *Void-check*

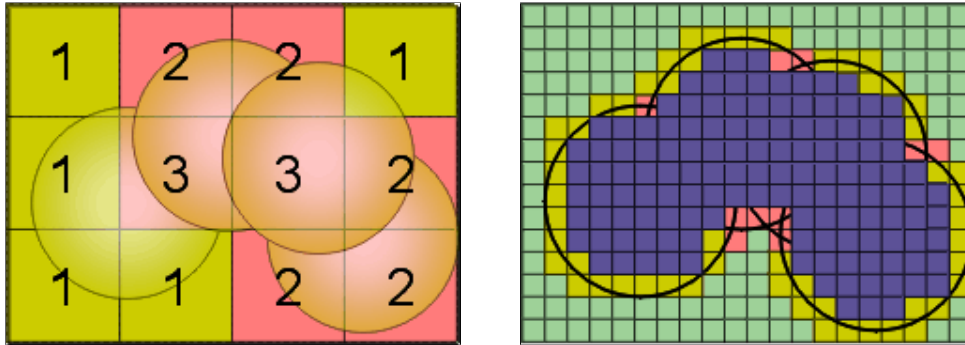
Eine Abschätzung der Größe der Slice-Würfel ergibt, dass diese Würfelchen üblicherweise *kleiner* sind als die Kugeln, die sich in der Packung befinden: Das o. g. Beispielsystem mit 8500 Atome hat vielleicht eine Kantenlänge von 5 nm; damit ergibt sich eine Kantenlänge der erzeugten 50 000 Slice-Würfel von etwa 130 pm — Der kleinste aus COMPASS ermittelte Hartkugel-Radius (für Wasserstoff<sup>50</sup>) beträgt 125 pm. Sein Durchmesser ist also bereits größer als die Kantenlänge des Mini-Würfels. Kommt der Radius eines zweiten Atoms als Penetrant (z. B. von O<sub>2</sub>-Sauerstoff: 152 pm) hinzu, steigt der Durchmesser der Hart-Kugeln im System auf mindestens 550 pm an. Dort hinein würden dann sogar schon mehrere der Slice-Würfel passen.

Es ist also zweckmäßig, *vor* der vollständigen Analyse eines jeden Slice-Würfels zu prüfen, ob der Würfel *komplett* im Inneren *einer* Kugel liegt (sog. *void-checking*, vgl. „void-mapping“ in [69]) — falls ja, enthält der Slice-Würfel ausschließlich besetztes Volumen; in den hier untersuchten Systemen ist dies für mindestens 90% aller Slice-Würfel der Fall.

Das Kriterium, ob ein Slice-Würfel vollständig im Inneren einer Kugel liegt, wird überprüft, indem der Abstand aller 8 Ecken des Würfels zu einem Kugelzentrum ermittelt

---

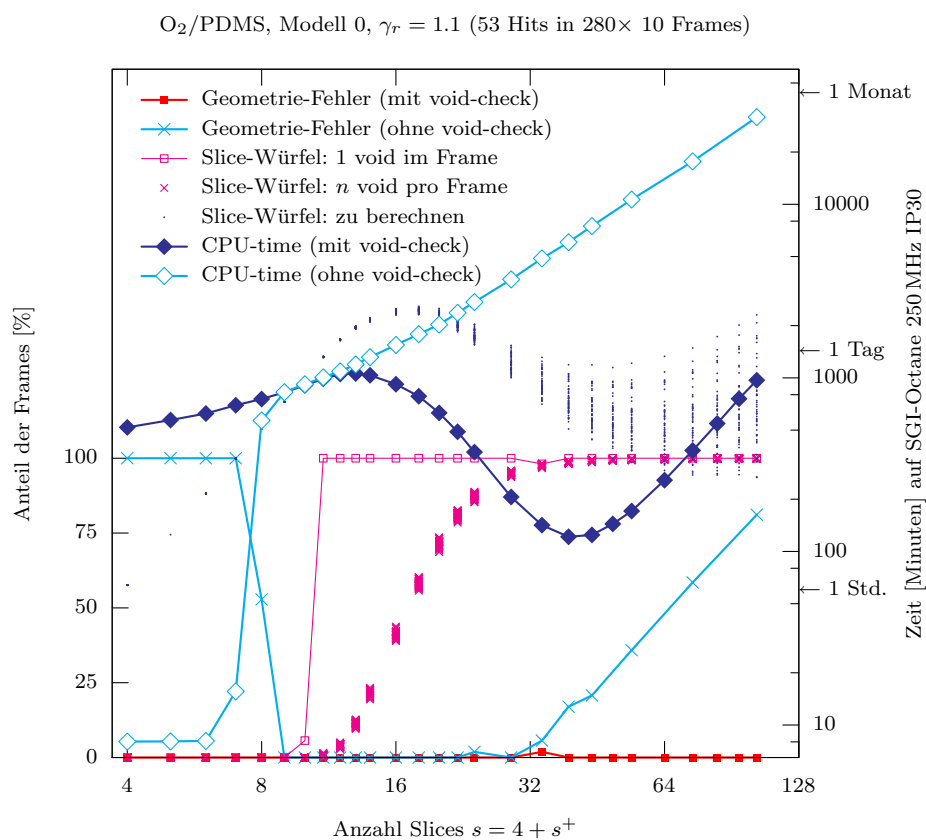
<sup>50</sup>Vergleiche [68]: van-der-Waals Radius von Wasserstoff = 120 pm.



**Abbildung 13:** Verringerung des Rechenaufwandes durch mehr Slices. Farb-Codierung der Slice-Würfel: blau = 100% belegt, grün = 0% belegt, gelb = nur eine Kugeloberfläche ( $\rightarrow$  schnell zu berechnen), rot = mehrere Kugeln überlappen.

wird: sind alle Abstände kleiner als der Radius der Kugel, so liegt der Slice-Würfel innen (falls mindestens eine Ecke außen liegt, wird mit der nächsten Kugel verglichen). Praktisch wird diese Untersuchung durchgeführt, wenn ohnehin alle Kugeln daraufhin untersucht werden, ob sie im oder in der Umgebung des Slice-Würfels liegen. Letztendlich muß nur noch das belegte Volumen von denjenigen Slice-Würfeln berechnet werden, die auf der Oberfläche der Kugeln liegen. Für den Fall, dass ein Slice-Würfel von *keiner* Kugel berührt wird, ist ebenfalls sofort bekannt, dass dieser Slice-Würfel ausschließlich aus zugänglichem Volumen besteht (also *kein* besetztes Volumen hat). In Abbildung 13 ist schematisch dargestellt, dass bei kleineren Slice-Würfeln im Verhältnis weniger der zeitaufwändigen Volumenberechnungen (rote oder gelbe Farbe) nach Dodd und Theodorou durchgeführt werden müssen; auch ist die Komplexität der noch zu berechnenden Slice-Würfel geringer geworden (weniger Kugeln sind enthalten). Der Gewinn an Rechenzeit wird jedoch bei *zu* kleinen Slice-Würfeln wieder aufgebraucht, weil die Zerlegung in Slice-Würfel dann überproportional viel Rechenzeit benötigt.

Die Rechenzeit des in Abbildung 14 gezeigten Beispiels ließ sich durch die Vorabsuche nach eingeschlossenen Slice-Würfeln um den Faktor 65 (!) beschleunigen (1.9 Stunden statt 5.2 Tage bei 40 Slices). Auch zwischen minimal 4 und 40 Slices ergibt noch eine deutliche Beschleunigung um den Faktor 4.5 (8.5 h  $\rightarrow$  1.9 h). Neben der benötigten



**Abbildung 14:** Einfluss der Größe der Slice-Würfel und des void-checkings: Weniger Geometrie-Fehler (cyan  $\times$ ) und kürzere Rechenzeit (blaue Rauten) durch mehr „void“ Slice-Würfel

CPU-Zeit und dem Anteil an aufgetretenen Fehlern ist in Abb. 14 auch die Anzahl der vollständig zu analysierenden Slice-Würfel angegeben (für jeden der 53 Hit-Frames einzeln). Das Endergebnis der Löslichkeitsberechnungen ist nicht dargestellt, da alle Werte im Rahmen der Rechengenauigkeit<sup>51</sup> identisch waren (soweit keine Geometriefehler aufgetreten sind).

Ein positiver Nebeneffekt des void-checking ist, dass vor allem diejenigen Slice-Würfel übergangen werden können, die aufgrund ihrer komplexen Zusammensetzung (viele

<sup>51</sup>Die Endergebnisse zeigen Abweichungen nach der vierten Dezimalstelle; Zwischenergebnisse stimmen bis zur achten Dezimale überein.

Kugeln) normalerweise Probleme bereiten würden (Tabellen zu klein oder externe Kugeln). Dieser Effekt ist ebenfalls Abbildung 14 im Bereich  $s > 30$  zu sehen.

Die Geometrie-Fehler für  $s < 9$  *ohne* void-check wurden durch eine zu kleine Tabelle in der Volumen-Berechnung verursacht. Da diese Fehler sehr schnell bemerkt wurden, ist hier auch kaum CPU-Zeit verbraucht worden. Die verursachende Tabelle wurde im Rahmen der Implementierung des void-checking vergrößert, so dass *mit* void-check dieser Fehler die Berechnungen nicht mehr stören konnte. (Nach Vergrößern der Tabellen gleicht die Laufzeit *ohne* void-check derjenigen *mit* void-check für  $s < 12$ .)

## 3.4 Praktischer Ablauf der Löslichkeits-Berechnung

### 3.4.1 Drei Phasen der Berechnung

Die komplette Berechnung der Löslichkeitswerte findet als post-processing *nach* der MC- oder MD-Simulation des untersuchten Systems statt. Die zugrunde liegende Simulation (quasi Phase Null) muss dabei gewissen Anforderungen genügen, damit sie zur Ermittlung des chemischen Exzesspotenzials sowie des Löslichkeitskoeffizienten genutzt werden kann: Die verwendete Simulationszelle muss ausreichend groß sein, um einen realistischen Ausschnitt des untersuchten Materials zu beschreiben; außerdem muss ein Penetrant-Molekül enthalten sein (genauere Details folgen im nächsten Abschnitt). Eine Trajektorie ist ein zeitlich vergrößertes Abbild der durchgeführten Simulation. Sie enthält eine Sammlung von Konfigurationen, die mit einem gewissen Abstand (z. B. alle 1000 Simulationsschritte) abgespeichert wurden. Für jede einzelne Konfiguration (sog. *Frames*) ist bekannt, an welcher Position im Raum sich jedes einzelne Atom der Simulationszelle befindet und in welcher Weise die Atome miteinander verbunden sind (Beschreibung der molekularen Struktur).

Die Ergebnisse der Simulation werden zunächst (Phase 1) mit Hilfe eines Überführungs-Scripts (Programmablaufplan in Abb. 15) aus der Trajektorie gelesen und in einem definierten Format (siehe Anhang A.6) als sog. `.struct(ure)`-Datei gespeichert. Die

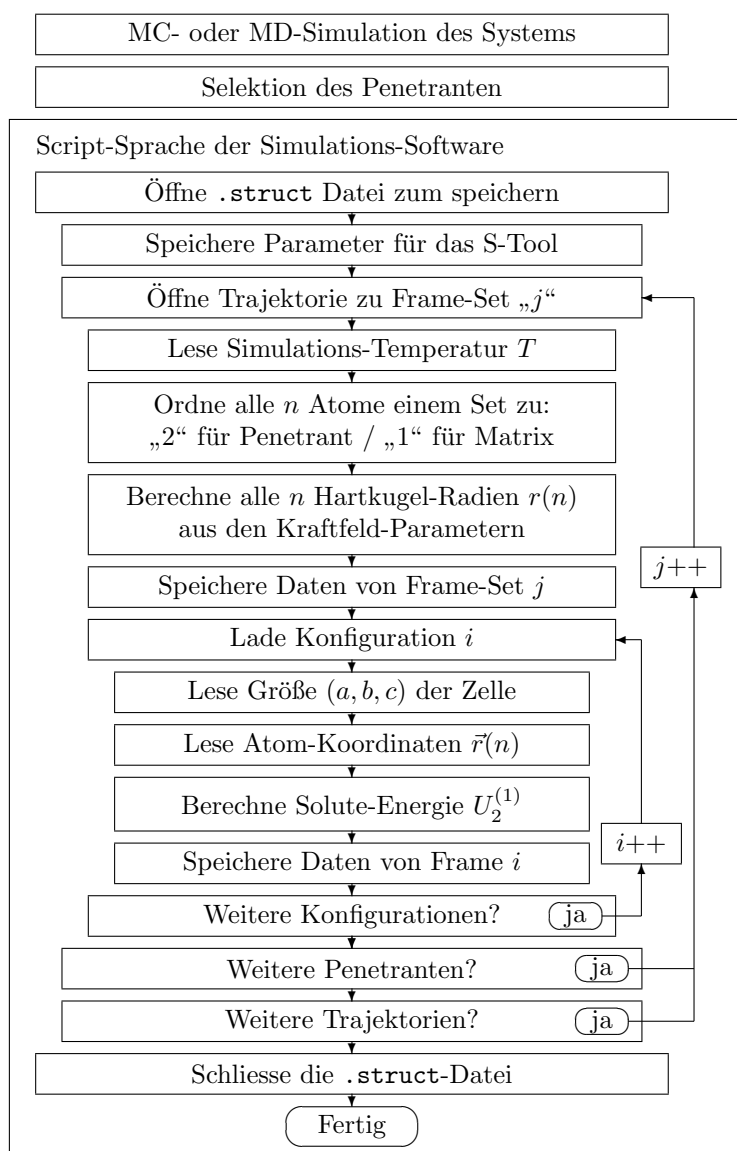


Abbildung 15: Ablauf des Überführungs-Scriptes

.struct-Datei enthält alle Eingabe-Daten sowie Steueranweisungen für das nachfolgende Rechenmodul „S-Tool“ (s. u.). Sie liegt als ASCII-Datei vor und kann daher bei Bedarf mit einem beliebigen Editor nachbearbeitet werden, um z. B. zusätzliche Parameter oder Befehle einzufügen. Das Script wurde in der Steuerungssprache der Simulationssoftware geschrieben, so dass es zum Einen Zugriff auf das (eventuell pro-

prietäre) Format<sup>52</sup> der Trajektorie hat und zum Anderen, aufgrund des dokumentierten, einfachen Formats der `.struct`-Datei, relativ leicht an andere Simulationssoftware angepasst werden kann (bzw. in deren Scriptsprache neu aufgesetzt werden kann). Bei der Erzeugung der `.struct`-Datei werden die Funktionen der Simulationssoftware genutzt, um die benötigten thermodynamische Informationen für die DPD Berechnung zu gewinnen: Solute-Energie und System-Temperatur.

Die eigentliche Hauptarbeit (Phase 2) leistet dann das compilierte Hauptprogramm „S-Tool“ (Programmablaufplan siehe Abb. 16 auf Seite 70), welches die `.struct`-Datei einliest und interpretiert. Es nutzt die enthaltenen Informationen, um die noch fehlenden, geometrischen Informationen für die DPD Gleichung zu berechnen. Abschließend führt es die Berechnung des chemischen Exzesspotenzials gemäß der DPD Gleichung (23) durch.

### 3.4.2 Notwendige Kenngrößen der Trajektorie

**Penetranten-Anzahl** Das simulierte System soll gemäß der ursprünglichen Herleitung der DPD Gleichung neben einer großen Menge an Lösungsmittel (Matrix, hier meist Polymer) genau *ein* gelöstes Teilchen (Penetrant) enthalten. Es ist jedoch akzeptabel, dass sich mehrere Penetrant-Moleküle in der Simulationszelle befinden, soweit die Zelle groß genug ist, so dass je zwei Penetranten gegenseitig nicht (oder nur kurzfristig) miteinander in Wechselwirkung treten. In diesem Fall wird für die Berechnung der Löslichkeit genau eines der Penetrant-Moleküle als *der* Penetrant ausgewählt und die restlichen Penetranten werden während der Analyse der Matrix zugerechnet — es liegt also formal ein, durch die zusätzlichen Penetrant-Moleküle, leicht gequollener<sup>53</sup> Zustand der Matrix vor. Der Vorteil dieser Herangehensweise liegt darin, dass die gleiche Konfiguration erneut, diesmal jedoch unter Wahl eines *anderen* Penetrant-Moleküls als *der*

---

<sup>52</sup>Im Falle der ACCELRYs-Simulationssoftware *Discover*, eine Kombination aus drei Dateien, in denen die Atomtypen und Anfangs-Koordinaten (`*.car`), die Bindungs-Struktur (`*.mdf`) und die zeitliche Entwicklung von Koordinaten und Geschwindigkeitsvektoren der Atome (`*.his`) festgehalten sind.

<sup>53</sup>Im Falle einer NVT Simulation bewirken die zusätzlichen Moleküle eine geringe Zunahme der Dichte.



Penetrant, ausgewertet werden kann. Mit einer Simulation werden also gleich mehrere „virtuelle“ Trajektorien erzeugt, was praktisch eine große Einsparung an Rechenzeit bedeutet.

⇒ Die praktische Erfahrung in der Anwendung der generalisierten DPD Methode (siehe 4.5.2) hat gezeigt, dass bei Packungsgrößen von  $\approx 6000$  Polymeratomen, wie sie üblicherweise in der Arbeitsgruppe verwendet werden, zumindest bis zu 10 mittelgroße Penetrant-Moleküle enthalten sein dürfen. Die berechneten Löslichkeitskoeffizienten eines solchen „M+10“ Systems unterscheiden sich nur im Rahmen der statistischen Streuung vom Erwartungswert aus zehn „M+1“ Systemen, die den Zustand der unendlichen Verdünnung formal besser repräsentieren.

Eine weitere Motivation, die Penetranten-Anzahl größer als Eins (oder Zehn) zu wählen liegt in der Möglichkeit, das chemische Exzesspotenzial auch für gequollene Systeme bei endlichem Druck zu berechnen; also nicht nur für den Grenzfall der unendlichen Verdünnung, sondern auch für beispielsweise glasige Polymere bei bis zu 50 Bar CO<sub>2</sub>-Druck. Die Zulässigkeit dieser „Betriebsart“ des S-Tools wird zur Zeit noch genauer untersucht.

**Frame-Abstand** Die Gesamtheit der einzelnen Konfigurationen (*Frames*) der Trajektorie soll einen möglichst repräsentativen Teil des Konfigurationsraums des Penetrant-Matrix Systems wiedergeben. Um dies zu erreichen, müssten theoretisch alle Konfigurationen „frisch“ erzeugt werden. Praktisch ist jedoch die Erzeugung einer realistischen Packung von Polymermolekülen alles Andere als trivial. Daher wird die sehr zeitaufwändige Erzeugung der Packung in der Regel nur einige wenige Male durchgeführt. Weitere Konfigurationen werden aus dieser Anfangskonfiguration mittels MC- oder MD-Simulation gewonnen und jeweils nach einer bestimmten Anzahl von Simulationsschritten als Frame in der Trajektorien-Datei abgespeichert. Bei dieser Vorgehensweise geht es primär darum, Veränderungen in der Umgebung eines bestimmten

Penetrantmoleküls zu bewirken und zu beobachten und weniger darum „größere“ globale Effekte im gesamten System zu erzielen.

⇒ Ein üblicher Zeitschritt<sup>54</sup> für detailliert-atomistische Polymer-MD-Simulationen beträgt 1.0 fs. Es hat sich gezeigt, dass Frames, etwa mit einem zeitlichen Abstand von 1 ps (also 1000 Simulationsschritten) gespeichert werden sollten. Bereits 250 bis 300 dieser Konfigurationen reichen in den meisten Fällen aus, um den Löslichkeitskoeffizienten abzuschätzen. Eine größere Anzahl von untersuchten Konfigurationen erhöht vor allem die Genauigkeit des berechneten chemischen Potenzials bzw. des Löslichkeitskoeffizienten.

### 3.4.3 Aufgaben des Überführungs-Scripts

Für das in dieser Arbeit entwickelte Überführungs-Script wurde die Sprache BTCL verwendet, die von der Simulations-Software *Discover* der Firma ACCELRYs verwendet wird. Bei BTCL handelt es sich um eine Erweiterung der von James Oosterhout [65] entwickelten Script-Sprache TCL durch die Firma BIOSYM (jetzt ACCELRYs), um die Kontrolle von MD-Simulationen zu ermöglichen.

Das Script gestattet das Einlesen der Trajektorien-Daten<sup>55</sup> und deren Überführen in ein definiertes Format (siehe Anhang A.6) für das nachfolgende S-Tool. Desweiteren berechnet das Script die entsprechenden Hartkugel-Radien aller Atome aus deren jeweiligem Atomtyp (Gleichungen (27) und (28)), sowie die Solute-Energie<sup>56</sup>  $U_2^{(1)}$  zwischen dem als Penetrant identifizierten Molekül und der restlichen Matrix (siehe A.5.2). (Die

<sup>54</sup>Dieser Zeitschritt von  $\partial 1 \text{ fs} = 1 \cdot 10^{-15} \text{ s}$  reicht aus, um die schnellsten Bewegungen im System mit ausreichender Genauigkeit aufzulösen. Beispielsweise die C–H-Valenz-Schwingungen, bei ca.  $3000 \text{ cm}^{-1}$ , was einer Schwingungsperiode von etwa  $1 \cdot 10^{-14} \text{ s}$  ( $=10 \text{ fs}$ ) entspricht; die Schwingung wird also in etwa 10 Zwischenschritte zerlegt.

<sup>55</sup>Kantenlängen  $a, b, c$  und Winkel  $\alpha, \beta, \gamma$  der PBC; Simulations-Temperatur  $T$ ; Koordinaten  $\vec{r}$  bzw.  $x, y$  und  $z$  aller Atome sowie deren Typ; Atom-Kategorie (Penetrant oder Matrix).

<sup>56</sup>Die Solute-Energie ist die Summe aller nicht-bindenden Wechselwirkungs-Energien zwischen jedem Atom des Penetranten und jedem Matrix-Atom, basierend auf dem jeweils gültigen Kraftfeld, welches während der Simulation Verwendung fand.

Simulations-Temperatur lässt sich aus Masse und Geschwindigkeit aller Atome berechnen; dies wurde jedoch nur zur Kontrolle gemacht, da für alle Simulationen die Temperatur explizit vorgegeben war und sich während der Simulationen nur im Rahmen des (ebenfalls vorgegebenen) Temperatur-Fensters von üblicherweise 10 K bewegte.)

Das Script besteht aus technisch zwei Komponenten: Eine *statische* Komponente stellt einige wenige zusätzliche TCL-Befehle (`Solubility_...`, siehe Seite 141 in Anhang A.5.1) zur Verfügung; sie wird am Anfang des User-Scripts geladen. Die zweite Komponente ist eben dies *User-Script*, welches jeweils an die Simulations-Trajektorie angepasst wird. Es verwendet die vorgenannten neuen TCL-Befehle, um die Trajektorien-Informationen in der `.struct`-Datei zu speichern. Ein exemplarisches User-Script ist in Anhang A.5.1 aufgelistet und dort detailliert kommentiert.

#### 3.4.4 Aufgaben des Hauptprogramms „S-Tool“

Das Hauptprogramm zur Löslichkeitsberechnung (das sogenannte *S-Tool*, siehe Abb. 16) wurde hauptsächlich<sup>57</sup> in FORTRAN77 programmiert und für Windows-PCs (x86), Apple-MacOS X (PowerPC) und SGI-Workstations (MIPS) compiliert. Da die Anpassung an unterschiedliche Simulations-Software im vorgenannten Überführungs-Script erfolgt, kann das *Executable* unverändert weiter benutzt werden, wenn man auf eine andere Simulations-Software wechselt; nur das Script muss angepasst werden bzw. dessen Funktionalität in der jeweiligen Script-Sprache der alternativen Simulations-Software neu implementiert werden.

Alle Steuerungs-Optionen des S-Tools werden aus der o. g. `.struct`-Datei gelesen. Das S-Tool erfüllt zwei Hauptaufgaben: 1.) Die Geometrie-Berechnung zur Ermittlung des Anteils an zugänglichem Volume  $\phi$ , sowie 2.) Die Zusammenführung aller Eingangswerte der DPD Gleichung zur Berechnung der Erwartungswerte und des daraus resultierenden chemischen Potentials  $\mu^{\text{ex},\infty}$  gemäß Gleichung (23) sowie des Löslichkeitskoeffizienten  $S$ . Beide Aufgaben werden quasi parallel (in der selben Schleife) durchgeführt,

---

<sup>57</sup>Es sind vier zusätzliche I/O-Routinen in C enthalten.

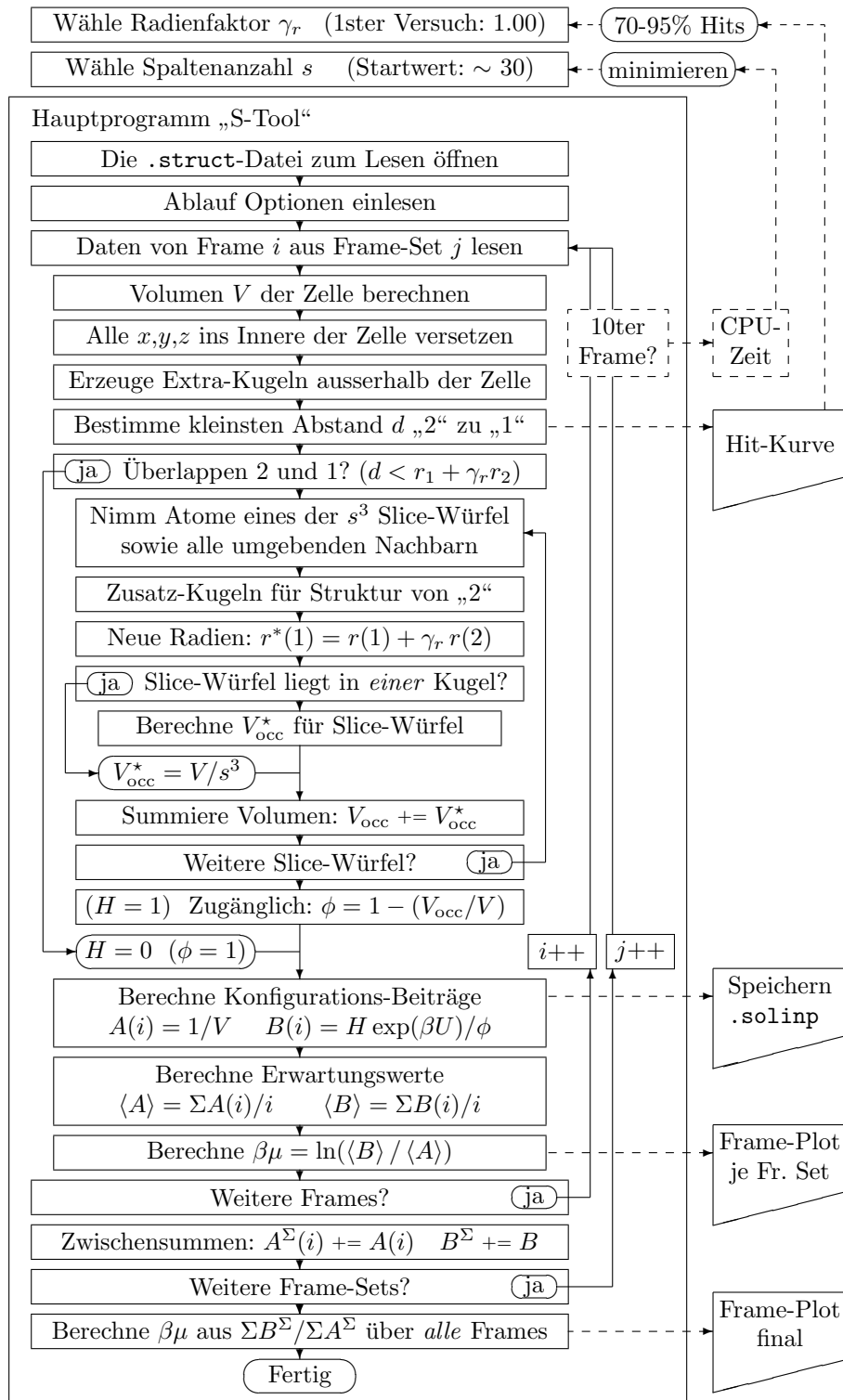


Abbildung 16: Ablauf des S-Tool Hauptprogrammes

wobei die Geometrieberechnung die zeitaufwändigere ist; daher werden die Geometrie-Ergebnisse in einer `.solinp`-Datei gespeichert, die strukturell der vorgenannten `.struct`-Datei ähnelt. Falls eine Berechnung unterbrochen wird, kann durch manuelles Zusammenfügen der Ergebnisse der bereits bearbeiteten Konfigurationen aus der `.solinp`-Datei mit dem unbearbeiteten Rest aus der `.struct`-Datei an der Stelle wieder angesetzt werden, an der die Bearbeitung zuvor unterbrochen wurde.

Die DPD Gleichung kann bei vorhandenen Eingangswerten schnell gelöst werden, so dass diese Berechnung jeweils sofort nach der Geometrie-Analyse jeder einzelnen Konfiguration vorgenommen wird. Der Verlauf der berechneten Löslichkeitskoeffizienten mit steigender Anzahl an untersuchten Konfigurationen wird mit dem sog. *Frame-Plot* dargestellt (siehe 4.2.1). Das Einzelergebnis berücksichtigt jeweils *ein* individuelles Frame Set, d. h. ein bestimmtes Packungsmodell mit einem Penetranten.

Die Eingangswerte der DPD Berechnung eines jeden untersuchten Frames werden zusätzlich in einer temporären Datei zwischengespeichert, und nach Geometrie-Analyse *aller* verfügbaren Frame Sets (Trajektorien bzw. Penetrantmoleküle) werden diese Werte erneut für eine Auswertung gemäß der DPD Gleichung herangezogen. Dies ergibt dann, berechnet über die Werte aller Konfigurationen, den endgültigen Erwartungswert für chemisches Exzesspotenzial und Löslichkeitskoeffizient (vgl. 4.2.3).

### 3.4.5 Details der Berechnung

**Vorbereitung** Es werden zunächst mittels der Simulations-Software die benötigten Trajektorien gemäß 3.4.2 erzeugt. Als nächstes wird das User-Script (siehe 3.4.3) an die erzeugte Trajektorie(n) angepasst — dies betrifft hauptsächlich die Identifizierung des Penetrant-Moleküls, sowie die Berücksichtigung mehrerer alternativer Simulationen des untersuchten Systems. An dieser Stelle werden auch bereits alle Steueranweisungen und Parameter für den nachfolgenden Aufruf des S-Tools gesetzt. Schließlich wird das Script zur Ausführung gebracht, um die `.struct`-Datei zu erhalten.

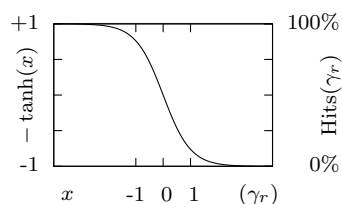
**Aufruf von S-Tool** Das S-Tool wird gestartet, um die zuvor erzeugte `.struct`-Datei zu bearbeiten. Hierbei sind zwei Parameter in der `.struct`-Datei frei wählbar. Diese beiden Parameter können bereits während der Vorbereitung auf erfahrungsgemäß sinnvolle Werte gesetzt worden sein oder sie werden (wie nachfolgend beschrieben) mittels einiger kürzerer Vorab-Untersuchungen (mit dem S-Tool) ermittelt.

**Penetrant-Radien Skalierungsfaktor  $\gamma_r$**  Den statistisch verlässlichsten Löslichkeitskoeffizienten erhält man, wenn möglichst wenige der erzeugten Konfigurationen aus der Berechnung des DPD Erwartungswertes herausgenommen werden. Nur ganz besonders extreme Konfigurationen mit sehr starker Annäherung des Penetranten an einzelne Matrix-Atome sollten durch die  $H$  Funktion ausgeschaltet werden. Diese Konfigurationen gehen üblicherweise mit einer sehr großen (abstoßenden) Solute-Energie einher und erlauben es nicht, den Penetranten durch eine harte Kugel zu ersetzen, um den fiktiven Zwischenzustand II (siehe Seite 29) im Rahmen des DPD Methode zu erreichen.

Ist ein sinnvoller Wert für  $\gamma_r$  nicht bekannt, so wird die `.struct`-Datei zunächst mit einem voraussichtlich zu großen Wert (z. B. 1.2) untersucht. Hierbei werden (fast) ausschließlich Überlappungen festgestellt — das S-Tool meldet das Ausmaß<sup>58</sup> der einzelnen Überlappungen. Daraus lässt sich für jeden Frame berechnen, ab welchem  $\gamma_r$ -Wert er keine Überlappung mehr zeigen würde (vgl. 4.2); solche Frames *ohne* Überlappung werden „Hit-Frame“ genannt. Der Anteil an Hit-Frames in Bezug auf die Anzahl aller untersuchten Frames wird als „Hit-Rate“ bezeichnet. Die Korrelation von Hit-Rate und Penetrant-Radien Skalierungsfaktor  $\gamma_r$  zeigt einen charakteristischen Verlauf ähnlich der negativen hyperbolischen Tangensfunktion (siehe Abb. 17). Dieser charakteristische Verlauf wird schließlich die *Hits-Kurve* genannt und ist in allen nachfolgend gezeigten Radien-Plots (z. B. in Abb. 22) enthalten. Aus der Hits-Kurve kann dann derjenige

---

<sup>58</sup>Die Überlappung beschreibt die Differenz von Kern-Kern-Abstand und Radien-Summe; negative Werte stehen für Überlappung. Der kleinste Wert (der größte negative Wert) wird ausgegeben.



**Abbildung 17:** Verlauf der negativen hyperbolischen Tangensfunktion.

Wert für  $\gamma_r$  abgelesen werden, der die gewünschte Hit-Rate von beispielsweise 75% ergibt. Alternativ kann  $\gamma_r^{100\%}$  abgelesen werden, von dem aus auf  $\gamma_r^M$  geschlossen werden kann.

Im Rahmen dieser Arbeit war es auch notwendig, die beschriebene Unabhängigkeit der Löslichkeit in einem gewissen Bereich vom gewählten Penetrant-Radius nachzuweisen. Daher wurde der Skalierungsfaktor  $\gamma_r$  in der `.struct`-Datei systematisch verändert, und das S-Tool wiederholt zur Berechnung mit den modifizierten Radien verwendet. Die Ergebnisse dieser sehr umfangreichen Untersuchungen werden in den sog. „Radien-Plots“ dargestellt (siehe hierzu auch Abschnitt 4.2.2).

**SlicesPlus**  $s^+$  Zur effizienten Nutzung der Rechenleistung der Computer muß ein günstiger Wert für die Anzahl  $s$  an Slices ermittelt werden (vgl. 3.3.5). Bei *viel zu wenigen* Slices enthalten die Slice-Würfel schlicht zu viele Atome und die Volumen-Berechnung scheitert an der großen Anzahl von Atom-Paarungen. Bei *zu vielen* Slices hingegen erzeugt die Buchführung über die Mitglieder der einzelnen Slice-Würfel einen größeren Rechenaufwand als die Einsparung durch die *sicher* belegten Slice-Würfel. Ein Optimum zwischen Buchführungs-Aufwand und Geometrie-Einsparung muss also gefunden werden.

Hierzu werden nur die ersten etwa 20 Konfigurationen der `.struct`-Datei untersucht. Von Hand (bzw. per Kommandozeilen-Script) wird hierbei die Anzahl an Slices verändert, um ein Minimum an Rechenzeit zu verbrauchen (s. Abb. 14). Es hat sich

gezeigt, dass dieses Optimum für die untersuchten Systeme (Abmessungen finden sich in Tabelle 1 auf Seite 75) bei etwa 24 Slices (für flüssiges Benzol) bzw. zwischen 35 und 40 Slices (für die Polymer Modelle) liegt; diese Anzahl entspricht jeweils einer Slice-Würfel-Größe von etwa 90 pm Kantenlänge. (Das kleinere Argon-Modell, mit Argon als einatomigem Penetranten wurde in nur 2 Slices ( $\rightarrow$  8 Slice-Würfel) geteilt, um die Berechnung möglich zu machen, ohne nach einem optimalen Wert zu suchen.)