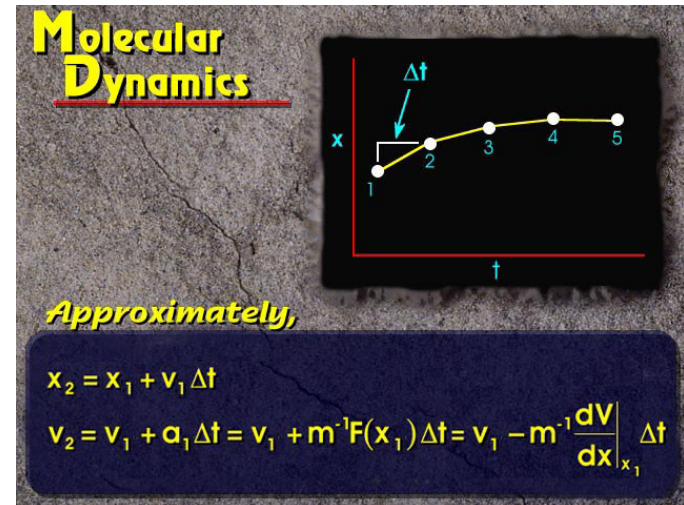


V5 Moleküldynamik

“Es gibt “nur” zwei Probleme bei MD-Simulationen:
das Kraftfeld und ausreichendes Sampling.”

Pflicht

Vorlesung 2	molekulare Kräfte
Vorlesung 3	Kraftfelder
Vorlesung 4	statistische Mechanik
Vorlesung 6	Ensembles und Sampling



Heute (V5):

- MD- Algorithmen
- Behandlung der langreichweitigen elektrostatischen WW
- Simulation bei konstanter Temperatur

Kür

Freie Energie-Rechnungen

Berechnung von Bindungsaffinitäten von Liganden in Enzymen

Moleküldynamik (MD)

MD-Simulation ist eine der Hauptmethoden für die Untersuchung biologischer Moleküle (Struktur, Dynamik und Thermodynamik) und ihrer Komplexe.

MD berechnet das zeitliche Verhalten eines molekularen Systems.

MD-Simulationen enthalten detaillierte Informationen über Fluktuationen und Konformationsänderungen von Proteinen und Nukleinsäuren (auf der entsprechenden Zeitskala. Aktuell durchführbare Simulationen decken den Bereich von 10^{-12} bis 10^{-6} sec, also pico- bis mikro-Sekunden ab)

MD-Simulationen werden auch unterstützend zur Strukturaufklärung in der Röntgenkristallographie und NMR-Messungen eingesetzt, sowie zur Nachoptimierung von gedockten Liganden im Rezeptor.

Anwendungsbereich von Newton-Dynamik

Newtonsche Physik verteilt die Energie gleichmäßig über alle vibrationelle Moden (Gleichverteilungs-Theorem), wogegen die Eigenzustände des harmonischen Oszillator jedoch gequantelte Energieabstände $h\nu$ haben.

MD Simulationen sind daher gut geeignet für Simulationen bei hohen Temperaturen (z.B. Raumtemperatur) und zum Sampling von Moden mit $\frac{h\nu}{k_B T} \ll 1$

Moden mit relativ großen Frequenzen wie Bindungsschwingungen (siehe Tabelle) sind bei Raumtemperatur jedoch fast immer im Grundzustand.

Daher ist klassische MD nicht gut geeignet um die Vibrationen solcher Bindungen und Bindungswinkeln zu simulieren. Hierzu sind (eigentlich) quantenmechanische Methoden

Vibrational mode	Wave number ($1/\lambda$) [cm^{-1}]	Frequency $\nu = c/\lambda$ [s^{-1}]	Ratio $h\nu/(k_B T)$
O–H stretch	3600	1.1×10^{14}	17
C–H stretch	3000	9.0×10^{13}	14
O–C–O asym. stretch	2400	7.2×10^{13}	12
C=O (carbonyl) stretch	1700	5.1×10^{13}	8
C–N stretch (amines)	1250	3.8×10^{13}	6
O–C–O bend	700	2.1×10^{13}	3

notwendig. Zusätzliches Problem sind die anharmonischen Effekte bei der Bindungsschwingung (Dehnen geht leichter als Stauchen).

Dynamische Prozesse in Biomolekülen

Lokale Bewegungen (0.01 bis 5 Å, 10^{-15} bis 10^{-1} s)

- Atomare Fluktuationen z.B. Bindungsschwingungen (sehr schnell 10^{-15} s)
- Bewegungen der Seitenketten von Aminosäuren, v.a. Rotationen, Umorientierung von Wassermolekülen und deren H-Brücken (relativ schnell, 10^{-12} bis 10^{-9} sec)
- Bewegung der *loops* von Proteinen (schnell, etwa 10^{-10} sec)

Bewegung starrer Körper (1 bis 10 Å, 10^{-9} bis 1s)

- Bewegung von Helices
- Domänenbewegung (hinge bending, wenige drehbare Bindungen involviert)
- Bewegung von Untereinheiten/Domänen

Weiträumige Bewegungen ($> 5\text{Å}$, 10^{-7} bis 10^4 s)

- Übergänge von Helix zu *coil* (langsam)
- Dissoziation/Assoziation Protein-Ligand (sehr langsam)
- Faltung und Entfaltung (teilweise sehr langsam, bis zu 10^4 s)

Finite Difference Methods

Die Intergration der Newtonschen Bewegungsgleichungen wird durch eine Summation über hinreichend kleine Zeitschritte Δt ersetzt.

Annahme: Alle Kräfte f die auf ein Atom wirken sind additiv (Vektoraddition).

Aus dieser Summe aller Kräfte ergibt sich eine Beschleunigung a des Atoms.

Zusammen mit der bereits vorhandenen Geschwindigkeit v und Position r des Atoms zum Zeitpunkt t wird die neue Geschwindigkeit und Position zum Zeitpunkt $t + \Delta t$ berechnet.

Geschwindigkeit: $v(t) = \frac{\partial r}{\partial t}$

Beschleunigung: $a(t) = \frac{\partial^2 r}{\partial t^2}$; Newton: $F = m \cdot a$

Dritte Ableitung: $b(t) = \frac{\partial^3 r}{\partial t^3}$

Annahme: Position, Geschwindigkeit, Beschleunigung usw. lassen sich als Taylorreihe approximieren (Approximation des jeweiligen Kurvenverlaufs):

$$r(t + \Delta t) = r(t) + \Delta t v(t) + \frac{1}{2} \Delta t^2 a(t) + \frac{1}{6} \Delta t^3 b(t) + \dots$$

Verlet-Algorithmus

Beim Verlet-Algorithmus benötigt man die Positionen r und Beschleunigungen a Zeitpunkt t , sowie die Positionen des vorhergehenden Schritts $r(t - \Delta t)$ um die Positionen am nächsten Schritt $r(t + \Delta t)$ zu berechnen:

$$r(t + \Delta t) = r(t) + \Delta t v(t) + \frac{1}{2} \Delta t^2 a(t) + \dots$$

$$r(t - \Delta t) = r(t) - \Delta t v(t) + \frac{1}{2} \Delta t^2 a(t) - \dots$$

Summation und Umformen dieser beiden Gleichungen liefert:

$$r(t + \Delta t) = 2r(t) - r(t - \Delta t) + \Delta t^2 a(t)$$

wobei die Geschwindigkeiten v nicht mehr auftauchen.

Diese lassen sich zwar ermitteln, jedoch mit gewissen Nachteilen (Zeitpunkt von Position und Geschwindigkeiten ist nicht synchron, siehe Leach, „Molecular Modelling“ Kapitel 6)

Es wurden deshalb Varianten des Verlet-Algorithmus entwickelt die die Positionen, Geschwindigkeiten und Beschleunigungen zum jeweils gleichen Zeitpunkt bestimmen.

Leap-frog Algorithmus

Leap-frog (“Bocksprung-“)
Algorithmus

mit

$$\mathbf{v}\left(t - \frac{\Delta t}{2}\right) \equiv \frac{\mathbf{r}(t) - \mathbf{r}(t - \Delta t)}{\Delta t}$$

$$\mathbf{v}\left(t + \frac{\Delta t}{2}\right) \equiv \frac{\mathbf{r}(t + \Delta t) - \mathbf{r}(t)}{\Delta t}$$

gilt

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \mathbf{v}\left(t + \frac{\Delta t}{2}\right)\Delta t$$

Für das *update* der
Geschwindigkeiten gilt:

$$\begin{aligned}\mathbf{v}\left(t + \frac{\Delta t}{2}\right) &= \frac{\mathbf{r}(t + \Delta t) - \mathbf{r}(t)}{\Delta t} \\ &= \mathbf{v}\left(t - \frac{\Delta t}{2}\right) - \frac{\mathbf{r}(t) - \mathbf{r}(t - \Delta t)}{\Delta t} + \frac{\mathbf{r}(t + \Delta t) - \mathbf{r}(t)}{\Delta t} \\ &= \mathbf{v}\left(t - \frac{\Delta t}{2}\right) + \Delta t \frac{\mathbf{F}(t)}{m}\end{aligned}$$

da aus Verlet – Algorithmus

$$\Delta t \frac{\mathbf{F}(t)}{m} = \frac{\mathbf{r}(t + \Delta t) - 2\mathbf{r}(t) + \mathbf{r}(t - \Delta t)}{\Delta t}$$

Ein einfaches Moleküldynamik-Programm

program md	einfaches MD program
call Init()	Initialisierung
t = 0	
while (t < tmax)	MD Schleife
call force(f, en)	berechne die Kräfte und Energien
call integrate(f, en)	integriere Bewegungsgleichungen
t = t + delta	
call sample	berechne Mittelwerte
endwhile	
stop	
end	

nach [Frenkel, Smit 1996]

Wiederholung: Molekülmechanik-Kraftfeld

$$\begin{aligned}
 E &= E_{stretch} + E_{bend} + E_{tors} + E_{vdW} + E_{ES} \\
 &= \sum_{bonds (ij)} \frac{k^{(ij)}}{2} (r_{ij} - r_0^{(ij)})^2 + \sum_{angles (ijk)} \frac{k^{(ijk)}}{2} (\phi_{ij} - \phi_0^{(ijk)})^2 \\
 &+ \sum_{torsions (ijkl)} \frac{k^{(ijkl)}}{2} (1 + \cos(n^{(ijkl)}\tau - \tau_0^{(ijkl)}))^2 \\
 &+ \sum_{pairs (ij)} \left(\frac{A_{(ij)}}{r_{ij}^{12}} - \frac{B_{(ij)}}{r_{ij}^6} \right) + \frac{1}{4\pi\epsilon\epsilon_0} \sum_{pairs (ij)} \frac{q_i q_j}{r_{ij}}
 \end{aligned}$$

} intra-molekular
} inter-molekular und intra-molekular

Historischer Hintergrund: immer am Rand des gerade noch „Machbaren“

1957/1959: Einführung der MD-Methode für die Wechselwirkung von harten Kugeln als Modell von Flüssigkeiten (Alder & Wainwright)

1964 erste Simulation mit realistischem Potential für flüssiges Argon (Rahman, 1964)

1974 erste MD-Simulation von Wasser (Stillinger and Rahman, 1974)

1977 erste MD-Simulation des Proteins BPTI (McCammon, *et al*, 1977).

1977 SHAKE-Algorithmus

1983 CHARMM, AMBER-Kraftfeld

1984 Freie-Energie-Rechnungen

1985 Car-Parrinello MD

1987 Gromos87-Kraftfeld

seit 1990 QM/MM Methode, z.B. für enzymatische Reaktionen

1992 Ewald-Methode

'90er Jahre: MD-Simulation auf parallelen Prozessoren → größere Systeme

Alder, B. J. and Wainwright, T. E. *J. Chem. Phys.* **27**, 1208 (1957), *J. Chem. Phys.* **31**, 459 (1959)

Rahman, A. *Phys. Rev.* **A136**, 405 (1964)

Stillinger, F. H. and Rahman, A. *J. Chem. Phys.* **60**, 1545 (1974)

McCammon, J. A., Gelin, B. R., and Karplus, M. *Nature (Lond.)* **267**, 585 (1977)

Initialisierung eines MD-Programms

```
subroutine init
sumv = 0
sumv2 = 0
for i=1, i<=npart, i++
  x(i) = lattice_pos(i)
  v(i) = (rand() - 0.5)
  sumv = sumv + v(i)
  sumv2 = sumv2 + v(i)**2
endfor
sum = sumv/npart
sumv2 = sumv2/npart
fs = sqrt(3*temp/sumv2)
for i=1, i<=npart, i++
  v(i) = (v(i) - sumv) * fs
  xm(i) = x(i) - v(i) * dt
endfor
return
end
```

Initialisierung

Summe der Geschwindigkeiten

Quadrat der Geschwindigkeiten

Schleife über Anzahl der Atome n_{part}

platziere Atom auf Gitter (oder lese Koordinate ein)

ordne Zufallsgeschwindigkeiten zu

Geschwindigkeit des center of mass (CMS)

kinetische Energie des CMS

Geschwindigkeit des CMS

mittlere quadrierte Geschwindigkeit

Skalierungsfaktor für Geschwindigkeiten – **warum?**

setze gewünschte kinetische Energie und setze

- Geschwindigkeit des Massenschwerpunkts = 0

Position im vorherigem Zeitschritt

- (für Integrationsalgorithmus)

Berechnung der Kräfte

```
subroutine force(f,en)
```

Initialisierung

```
en = 0
```

setze Energie auf 0

```
for i=1, i<=npart, i++
```

```
  f(i) = 0
```

setze Kräfte auf 0

```
endfor
```

```
for i=1, i<= npart-1, i++
```

```
  for j=i+1, j<= npart, j++
```

Schleife über alle Paare $j > i$

```
    xr=x(i)-x(j)
```

berechne Distanz zwischen Teilchen i und j

```
    xr=xr-box*int(xr/box)
```

periodische Randbedingungen

```
    r2=xr*xr
```

quadratischer Abstand

```
    if (r2 < rc2) then
```

überprüfe, ob $\text{Distanz}^2 < \text{cut-off}^2$ (erübrigt $\text{sqrt}()$)

```
      r2i=1/r2
```

hier nur Lennard-Jones Potential der Form

```
      r6i=r2i**3
```

$4 \epsilon \left(\frac{A}{r^{12}} - \frac{B}{r^6} \right)$

```
      ff=48*eps*r2i*r6i*(r6i-0.5)
```

Ableitung

```
      f(i)=f(i)+ff*xr
```

Update der Kräfte

```
      f(j)=f(j)-ff*xr
```

```
      en=en+4*r6i*(r6i-1)-ecut
```

Update der Energie

```
    endif
```

```
  endfor
```

```
endfor
```

```
return
```

```
end
```

nach [Frenkel, Smit 1996]

Ableitungen und Kräfte

Bindung zwischen Atom a und b: $\vec{f}_a = -2k(r - r_0) \cdot \vec{u}$ mit $\vec{u} = \text{norm}(ba) = \frac{\vec{ba}}{|ba|}$ $\vec{f}_b = -\vec{f}_a$

wobei u der Einheitsvektor (Länge = 1) von Atom b nach a ist.

Winkel zwischen a, b und c: $f_a = -2k(\theta - \theta_0)/|ab| \cdot \vec{p}_a$ mit $\vec{p}_a = \text{norm}(ba \times (ba \times bc))$

$$f_c = -2k(\theta - \theta_0)/|bc| \cdot \vec{p}_c \quad \text{mit } \vec{p}_c = \text{norm}(cb \times (ba \times bc)) \quad f_b = -f_a - f_c$$

Diederwinkel zwischen a, b, c und d: $f_a = \frac{0.5}{|ab| \sin(abc)} (V \sin(\varphi)) \cdot p_1$ mit $p_1 = \text{norm}(ba \times bc)$

$$f_c = \frac{1}{|oc|^2} t_c \times oc \quad \text{mit } o \text{ der Mitte zwischen b und c, und } t_c = -(oc \times f_d + 0.5 cd \times f_d + 0.5 ba \times f_a)$$

$$f_d = \frac{0.5}{|ab| \sin(bcd)} (V \sin(\varphi)) \cdot p_2 \quad \text{mit } p_2 = \text{norm}(cd \times cb) \quad f_b = -f_a - f_c - f_d$$

Lennard-Jones Potential zwischen a und b:

$$f_a = \frac{24\varepsilon}{r} \left[2 \frac{A}{r^{12}} - \frac{B}{r^6} \right] \cdot \vec{u} \quad \text{mit } u = \text{norm}(ab) \quad f_b = -f_a$$

Hinweis: Die Richtung einer Kraft ist durch ihren Vektor gegeben.

Integriere die Bewegungsgleichungen

```
subroutine integrate(f,en)      integriere Bewegungsgleichungen
sumv=0
sumv2=0
for i=1, i<= npart, i++      MD Schleife
  xx=2*x(i)-xm(i)+delta**2*f(i)  Verlet-Algorithmus
  v(i)=(xx-xm(i))/(2*delta)      Geschwindigkeit von Teilchen i
  sumv=sumv+v(i)                 Geschwindigkeit des CMS
  sumv2=0.5*mass(i)*sumv2*v(i)**2  totale kinetische Energie
  xm(i)=xx                       update der Koordinaten
endfor
temp=sumv2/(3*npart)           aktuelle Temperatur
etot=(en+sumv2)/(2*npart)      Gesamtenergie pro Teilchen
return
end
```

periodische Randbedingungen (PBC)

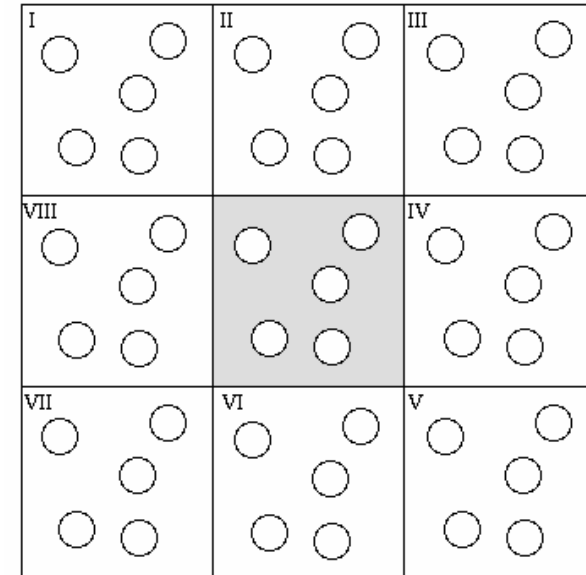
Verwendung von PBC (periodic boundary conditions) erlaubt es, Simulationen mit relativ kleiner Teilchenzahl durchzuführen, wobei die Teilchen Kräfte so wie in Lösung erfahren.

Die Koordinaten der „**Kopien**“ gehen aus denen der Teilchen in der zentralen Box durch einfache **Translationen** hervor.

Kräfte auf die zentralen Teilchen werden durch Wechselwirkungen innerhalb der Box sowie mit den Nachbarboxen berechnet.

Der **cut-off** wird so gewählt, daß ein zentrales Teilchen nicht gleichzeitig mit einem anderen zentralen Teilchen und dessen Kopie wechselwirkt.

Für eine kubische Box: $\text{cut-off} < \text{box}/2$



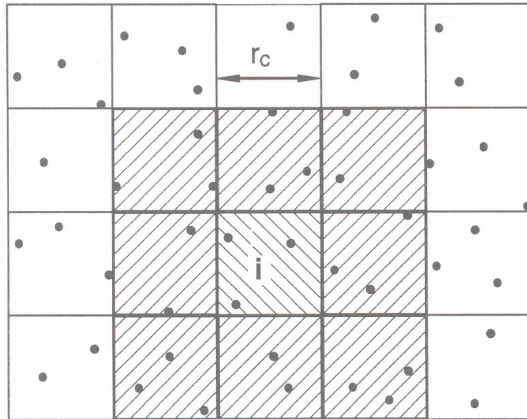
Bsp. 2-dimensionale Box

Zentrale Box ist von 8 Nachbarn umgeben.

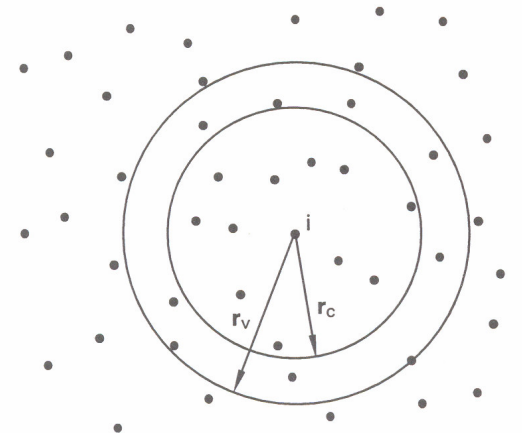
nach [Frenkel, Smit 1996]

Nachbarlisten bringen Speedup

Zell-Liste



Verlet-Liste



[Frenkel, Smit 1996]

Es gibt $N \cdot (N-1)$ Paar-Wechselwirkungen. Nicht alle müssen bei jedem Iterationsschritt berechnet werden! Spare Rechenzeit durch Nachbarliste, die Buch führt, welche Teilchen in benachbarten Zellen liegen (links) bzw. in äußerer Kugelschale (rechts).

Bei Schritt $i+1$ braucht nur überprüft zu werden, ob Teilchen aus diesen Zellen nun innerhalb des cut-off liegen. Skaliert mit N .

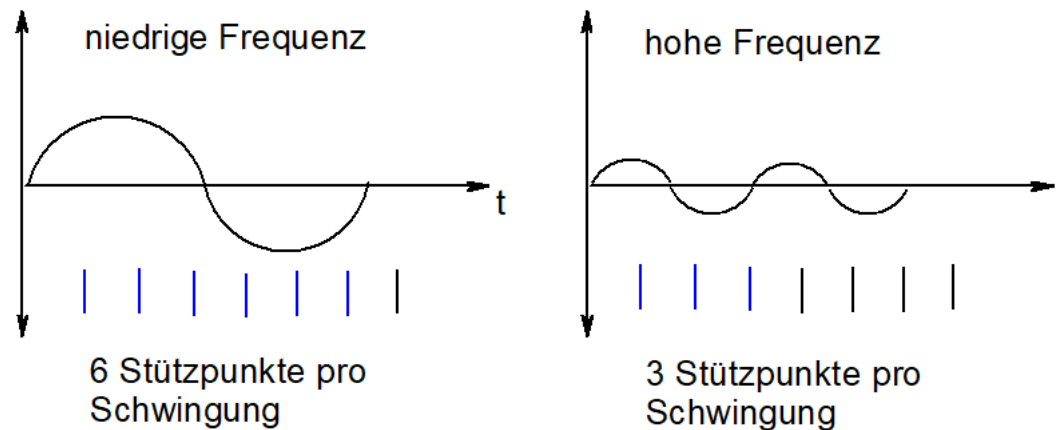
Alle n Schritte ($n \approx 10$) ist Neuberechnung der Nachbarliste nötig. Skaliert mit N^2 .

Algorithmen für MD (I)

Leap-frog und **Verlet**-Algorithmen sind am gebräuchlichsten in typischen MD-Simulationspaketen. Sie benötigen lediglich die **ersten Ableitung** der Energie nach den Teilchenkoordinaten, also die Kräfte auf die Atome.

In Simulationen im NVE-Ensemble (also bei konstanter Teilchenzahl N , konstantem Volumen V und konstanter Energie E) bleibt die Energie bei Verwendung von Simulationszeitschritten $\Delta t \leq 0.5$ fs erhalten. Ansonsten werden die Ungenauigkeiten bei der Erfassung der Bindungsschwingungen zu groß und die Simulation wird instabil.

Ursache sind die hochfrequenten Schwingungen (z.B. C-H, O-H, N-H), bei denen man prinzipbedingt weniger Stützpunkte pro Zeitschritt erfasst.



Algorithmen für MD (II)

Durch “Einfrieren” solcher Bindungslängen (mittels des “SHAKE” oder “LINCS” Algorithmus) kann man sogar Zeitschritte bis $\Delta t \leq 2.0$ fs verwenden.

Forester & Smith *J. Comput. Chem.* **19** (1998) 102–111.

Hess, Bekker, Berendsen & Fraaije *J. Comput. Chem.* **18** (1997) 1463–1472.

Es existieren kompliziertere Algorithmen, die **höhere Ableitungen** der Energie verwenden. Diese Algorithmen erlauben es, längere Zeitschritte bis ca. $\Delta t \leq 5.0$ fs zu verwenden.

Allerdings ist der **Aufwand**, die höheren Ableitungen zu berechnen, gewöhnlich grösser als die Einsparung an Schritten. Deshalb wird vor allem der effektive LINCS Algorithmus angewandt.

Aufgrund der mikroskopischen Reversibilität lässt sich auch rückwärts in der Zeit simulieren (was geschah vor dem Startpunkt der Simulation?)

Moleküldynamik

MD-Simulationen haben mehrere Aufgaben

- * Generierung von Konformationen,
Suche im Konformationsraum (lokal im Rahmen der vorgegebenen
Einschränkungen: Barrierenhöhe, Temperatur)
- * Dynamische Simulation von Systemen im Gleichgewicht
z.B. Verfeinerung von gedockten Enzym-Ligand Komplexen
(wie passt sich die Bindungstasche an, *induced fit*)
- * Nicht-Gleichgewichts-Simulationen, z.B. durch Anlegen von äußeren
Kräften, Potentialen, oder Ionenkonzentrationen
(Durchgang durch Membrankanäle, Änderung des pH-Werts ohne
die Anzahl der H_3O^+ Teilchen zu verändern)
→ erlaubt auch Situationen die experimentell nicht machbar sind.

Langreichweitige elektrostatische Wechselwirkung

Empirische Kraftfelder beschreiben die nichtgebundene Wechselwirkung meistens durch 2 Terme:

- ein Lennard-Jones-Potential $C_{12}/r^{12} - C_6/r^6$
- die Coulombsche Wechselwirkung aller Teilchen miteinander mittels ihrer Punktladungen q_i

$$U = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \frac{q_i q_j}{r_{i,j}}$$

Der Lennard-Jones Term **fällt** sehr **schnell** mit zunehmender Entfernung **ab** und kann guten Gewissens für $r \geq 1$ nm vernachlässigt werden.

Die elektrostatische Wechselwirkung fällt jedoch nur mit $1/r$ ab.

Die Verwendung einer Abschneidefunktion (*cut-off*) führt daher oft zu Artefakten, wie etwa (unrealistischem) Temperaturanstieg (siehe folgende Folie).

Cut-off Artefakte

- **Aufheizung des Systems** durch Austritt dipolarer Gruppen mit Vorzugsrichtung und Wiedereintritt mit beliebiger Orientierung (unrealistische Entropiezunahme)

- **strukturelle + dynamische Abweichung von experimentellen Daten!**

Zu große "Ordnung" des Systems, da jede *cut-off*-Sphäre quasi von Vakuum umgeben ist. Probleme bei Equilibrierung von kritischen Systemen, die z.B. viele polare Gruppen enthalten, deren Ladungen in der Lösung durch Gegenionen abgesättigt werden, wie DNA-Lösungen oder Biomembranen.

In MD-Simulationen von DNA mit *cut-off* wird z.B. während Simulation im Bereich von Nanosekunden keine stabile Konformation gefunden, daher große Abweichung von Kristallstrukturen.

- prinzipielles "**Problem**" bei Berücksichtigung der langreichweiten Elektrostatik: die empirischen Kraftfelder wurden ursprünglich in den '80er und '90er Jahren alle mittels Simulationen mit *cut-off* geeicht. Die Übereinstimmung beispielsweise der Diffusionskonstante von Wasser mit dem Experiment wird daher durch Einschalten der langreichweitigen Elektrostatik nicht besser, sondern schlechter.

Ewald-Summe

Bei dieser Methode stellt man sich das System unendlich weit in drei Dimensionen fortgesetzt, also periodisch, vor. Die Summationsmethode der gesamten Wechselwirkung in Kristallen wurde in den 1920er Jahren von Ewald entwickelt. (siehe A. Leach, Molecular Modelling, 2nd Ed., Chapter 6)

Der entscheidende Schritt ist, den kritischen Term r^{-1} in zwei rasch konvergierende Terme zu zerlegen.

76

A.Y. Toukmaji, J.A. Board Jr. / *Computer Physics Communications* 95 (1996) 73–92

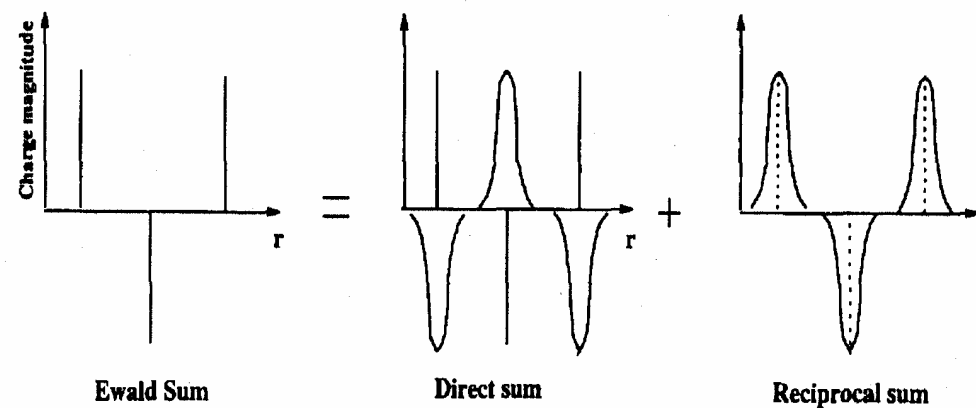


Fig. 2. The Ewald sum components of a one-dimensional point-charge system. The vertical lines are (+/-) unit charges, and the Gaussians are also normalized to unity.

Der linke Term konvergiert schnell im reellen Raum, der rechte schnell im reziproken Raum

Langreichweitige Elektrostatik

Es gibt weitere Techniken (Details siehe Leach, Molecular Modelling, Chapter 6)

- **Particle-Mesh-Ewald**, eine vor allem für größere Systeme sehr effiziente Variante der ursprünglichen Ewald-Summe
- **Fast-Multipole-Methode (auch Cell-Multipole-Methode genannt)**
- **Reaktionsfeld-Methoden**, nur für isotrope Systeme wie etwa Flüssigkeiten

Es stehen damit eine Reihe von zuverlässigen Methoden zur Berücksichtigung langreichweitiger elektrostatischer Effekte in Simulationen zur Verfügung.

Wichtig:

Durch Verwendung einer künstlichen Periodizität werden die strukturellen, dynamischen und vermutlich auch die thermodynamischen Eigenschaften der Systeme verändert! (z.B. Kopien von Membranen in unmittelbarer Nähe)
Dies muß jedoch nicht von Nachteil sein, sofern man die relative Größe dieser Effekte kennt und evtl. bei der (Neu-) Parametrisierung von Kraftfeldern berücksichtigt. → zahlreiche Parametervarianten bei gängigen Kraftfeldern

NVE Bedingung

Bislang wurden die Zustände des klassischen Systems unter den Bedingungen

N (Teilchenzahl) = konstant

V (Volumen) = konstant

E (Gesamtenergie) = konstant

simuliert.

Wenn man annimmt, daß zeitliche Mittelwerte = Ensemblemittelwerte sind, dann entsprechen Mittelwerte über eine konventionelle MD-Simulationen den Ensemble-Mittelwerten eines *mikrokanonischen Ensembles*.

Jedoch ist es oft wünschenswert, Simulationen in anderen Ensembles durchzuführen, z.B. NVT oder NPT , in denen die Temperatur T und/oder der Druck p konstant gehalten werden.

Bedingung konstanter Temperatur: Kopplung an Wärmebad

Wenn man ein System mit einem großen Wärmereservoir koppelt, bekommt das System eine gleichbleibende **Temperatur**.

D.h. die Anzahl der Zustände bei Energie E gehorcht nun der **Boltzmann-Verteilung**, und auch die Verteilung der **Geschwindigkeiten** $P(p)$ gehorchen für ein klassisches System der **Maxwell-Boltzmann-Verteilung**:

$$P(p) = \left(\frac{\beta}{2\pi m} \right)^{3/2} \exp \left[\frac{-\beta p^2}{2m} \right]$$

Das heisst, es gibt langsame und schnelle Lösungsmittelmoleküle entsprechend einer Boltzmann-Verteilung der Geschwindigkeiten.

Grundsätze

In einem klassischen Vielteilchensystem ist die **Energie** über all diejenigen Freiheitsgrade N_f **gleichverteilt**, die quadratisch in den Hamiltonian (Energiefunktion) des Systems eingehen, also z.B. die Geschwindigkeiten.

Für die mittlere kinetische Energie pro Freiheitsgrad gilt:

$$\left\langle \frac{1}{2} m v_i^2 \right\rangle = \frac{1}{2} k_B T$$

Die Klammer $\langle \dots \rangle$ bedeutet eine zeitliche Mittelung über die Konfigurationen des Systems z.B. während einer MD-Simulation.

In einer MD-Simulation kann man diese Beziehung bequem zur Berechnung der Temperatur T des Systems zum Zeitpunkt t heranziehen:

$$T(t) = \sum_{i=1}^N \frac{m_i v_i^2(t)}{k_B N_f}$$

D.h. die aktuelle Temperatur des Systems wird aus den Geschwindigkeiten der Teilchen berechnet (und fluktuiert deshalb etwas).

Andersen-Thermostat

Zu bestimmten Intervallen wird die Geschwindigkeit eines zufällig ausgewählten Teilchens neu aus einer Maxwell-Boltzmann-Verteilung der Geschwindigkeiten

$$P(\mathbf{p}) = \left(\frac{kT}{2\pi m} \right)^{3/2} \exp \left[\frac{-kT \mathbf{p}^2}{2m} \right]$$

zugeteilt, entsprechend einer stochastischen Kollision dieses Teilchens mit einem Solvensmolekül eines imaginären Wärmebads.

Problem: die stochastischen Kollisionen stören die dynamischen Eigenschaften in unrealistischer Weise. Die Geschwindigkeits-Autokorrelationsfunktion fällt zu schnell ab. Dieser Thermostat ist nicht geeignet um dynamische Eigenschaften wie Diffusionskoeffizienten zu simulieren. Alle statischen Eigenschaften werden aber korrekt berechnet.

Andere Verfahren (Nosé-Hoover) erlauben es, sowohl thermodynamische wie dynamische Eigenschaften methodisch sauber zu berechnen.

Protokoll für Moleküldynamik-Simulationen

generiere Koordinaten



minimiere Struktur (Energie Minimierung)



ordne (zufällige) Anfangsgeschwindigkeiten zu



Aufwärmphase (MD)



Equilibrierung (MD)

Skaliere
Geschwindigkeiten

Temperatur OK?

Nein

Ja



Produktionslauf (MD)



Analyse (der Trajektorien)

Analyse von Trajektorien

Mittlere Energie

$$\langle E \rangle = \frac{1}{N} \sum_{i=1}^N E_i$$

RMS-Unterschied zwischen
2 Strukturen

$$RMS = \left\langle \left(r_i^\alpha - r_i^\beta \right)^2 \right\rangle^{1/2} = \sqrt{\frac{1}{N_i} \left(r_i^\alpha - r_i^\beta \right)^2}$$

RMS-Fluktuationen

$$RMS_i^{fluct} = \sqrt{\frac{1}{N_f} \sum_f \left(r_i^f - r_i^{ave} \right)^2}$$

(diese hängen mit den
kristallographischen B-
Faktoren zusammen)

$$B_i = \frac{8}{3} \pi^2 \left(RMS_i^{fluct} \right)^2$$

Zusammenfassung

MD-Simulation ist eine etablierte und mittlerweile sehr robuste Simulationstechnik. Dazu tragen auch verbesserte bzw. re-parametrisierte Kraftfelder bei.

Ständig **neue Anwendungsgebiete** (z.B. größere und realistischere Systeme)

- (a) durch Beschleunigung der Algorithmen,
- (b) durch Zunahme der Prozessorgeschwindigkeit,
- (c) durch Parallelisierung der Programme/Verwendung von GPUs.

Ausgangspunkt ist gewöhnlich Röntgenkristallstruktur eines Biomoleküls oder eine modellierte Struktur die vorher energetisch optimiert wurden. (Warum?)

Wichtig sind

- sorgfältige Konstruktion der Startkonfiguration
- sinnvolle Wahl von Kraftfeld und Simulationsbedingungen
- sorgfältige Equilibrierung des Systems
- analysiere nur Eigenschaften des Systems für die das Sampling OK ist.