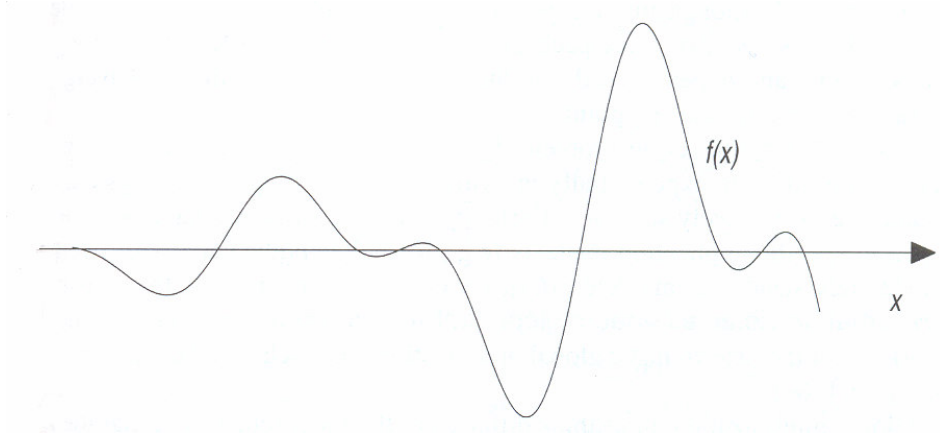
# 4. Vorlesung

Welche Minima gibt es auf der Energiehyperfläche?

Wie findet man diese effizient?

- Lokale und Globale Optimierung
- Sampling von Energiehyperflächen
- Monte Carlo / Metropolis-Algorithmus
- Statistische Mechanik
- Entropie, Phasenraum, Ensembles

# Algorithmen zur Energieminimierung




**Lokales Minimum**:

Für einen Vektor **x** mit *n* Komponenten $\{x_i\}$ lautet das Minimierungsproblem

$\min_{\mathbf{x}} \{ f(\mathbf{x}) \}$, $\mathbf{x} \in D \subset \Re^n$. Dann ist x ein lokales Minimum und

$f(\mathbf{x}) < f(\mathbf{y}) \;\; \forall\, \mathbf{y} \in D \subset \Re^n, \mathbf{y} \neq \mathbf{x}$

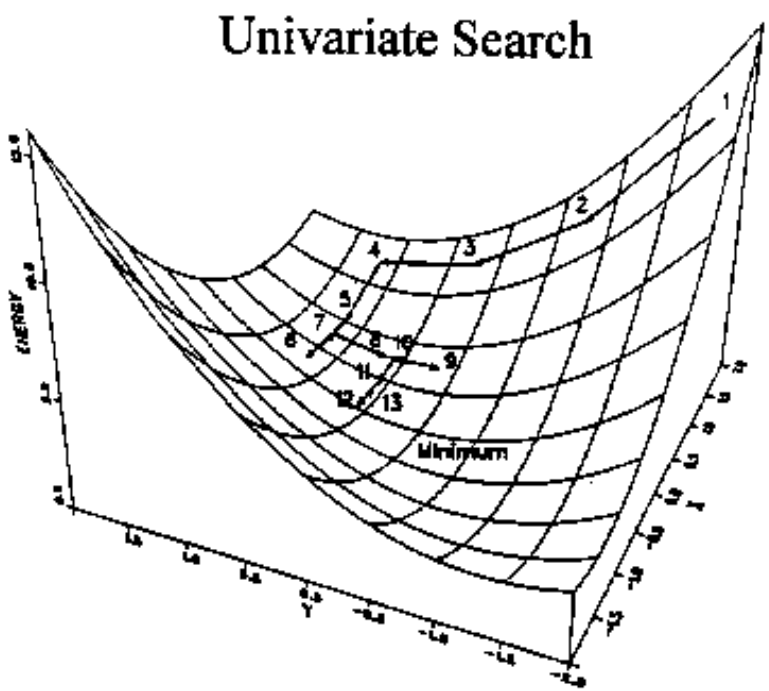
**Globales Minimum**:

$\mathbf{x} \subset \Re^n$ ist ein globales Minimum $\Leftrightarrow \;\; f(\mathbf{x}) < f(\mathbf{y}) \;\; \forall\, \mathbf{y} \subset \Re^n \neq \mathbf{x}$

[Schlick 10.2.4]

# Methode, die nur die Energiewerte verwendet ...

- Am **einfachsten** zu **implementieren**.
- Läuft in eine Richtung bis die Energie ansteigt, dreht dann um die Suchrichtung um 90º, etc.
- ist am **wenigsten effizient**
  - viele Schritte notwendig
  - Schritte sind nicht gerichtet
- wird daher selten verwendet.



# Grundlegende algorithmische Konzepte

Die grundlegende Struktur von iterativen, lokalen Optimierungsalgorithmen ist die des “greedy descent” (am schnellsten bergab).

D.h. ausgehend von einem Startpunkt $\mathbf{x}_0$ wird eine Sequenz von Schritten $\{\mathbf{x}_k\}$ erzeugt, wobei bei jeder Iteration versucht wird, den Wert der Zielfunktion $f(\mathbf{x})$ zu verringern.

Es gibt 2 Klassen von Algorithmen: **line-search** und **trust-region**.

Suche entlang einer Richtung derer Minimum

approximiere einen bestimmten Bereich

Beide sind weitverbreitet und sind Bestandteile von Optimierungsmethoden, die von jedem Startpunkt aus die Konvergenz zu einem lokalen Minimum garantieren. Beide Methoden sind gleichermaßen empfehlenswert.

# Line-Search basierender Descent Algorithmus [A1]

Von einem gegebenen Anfangspunkt $\boldsymbol{x}_0$ aus, führe für $k$ = 0, 1, 2, … die folgenden Schritte so lange aus bis Konvergenz eintritt.

(1) Überprüfe $\boldsymbol{x}_k$ auf Konvergenz (Abweichung unterhalb eines Grenzwerts)

(2) Berechne eine Abstiegs-Richtung $\boldsymbol{p}_k$

(3) Bestimme eine Schrittweite $\lambda_k$ durch eine 1-dimensionale Suche so dass für den neuen Positionsvektor $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \lambda_k \boldsymbol{p}_k$ und den entsprechenden Gradient $\boldsymbol{g}_{k+1}$ gilt:

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) + \alpha\lambda_k \mathbf{g}_k^T \mathbf{p}_k$$ (“ausreichende Abnahme”)

und

$$\left|\mathbf{g}_{k+1}^T \mathbf{p}_k\right| \leq \beta \left|\mathbf{g}_k^T \mathbf{p}_k\right| \qquad mit\ \ 0 < \alpha < \beta < 1$$ (“ausreichende Verringerung der gerichteten Ableitung”)

z.B. $\alpha = 10^{-4}$, $\beta = 0.9$ in Newton-Methoden

(4) Setze $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \lambda_k \boldsymbol{p}_k$ und $k \rightarrow k + 1$ und gehe zurück nach (1)

# [A1] (2): Richtung der Abnahme

Eine solche Richtung $\boldsymbol{p}_k$ ist eine Richtung, entlang derer die Funktion *f* lokal abnimmt.

Solch einen Vektor kann man durch $\mathbf{g}_k^T \mathbf{p}_k < 0$ definieren.

Für genügend kleine λ gilt dann (mit Tayler-Entwicklung um **x**):

$$f(\mathbf{x}+\lambda\mathbf{p}) - f(\mathbf{x}) = \lambda \mathbf{g}(\mathbf{x})^T \mathbf{p} + \frac{\lambda^2}{2} \mathbf{p}^T \mathbf{H}(\mathbf{x}) \mathbf{p}$$

$$\approx \lambda \mathbf{g}(\mathbf{x})^T \mathbf{p} < 0$$

Die Idee einer Tayler-Entwicklung ist, daß jede beliebiger Kurvenverlauf lokal um x durch eine Polynominalfunktion angenähert werden kann.
Im einfachsten Fall benutzt man dazu den Funktionswert f an der Stelle x und die erste Ableitung g an der Stelle x (Gradient oder Tangente).

# Steepest Descent (SD)

Die einfachste Möglichkeit, eine gültige Abnahme-Richtung vorzugeben, ist zu setzen:

$$\mathbf{p}_k = -\mathbf{g}_k$$

Damit ist automatisch erfüllt

$$\mathbf{g}_k^T \mathbf{p}_k = -\mathbf{g}_k^T \mathbf{g}_k$$

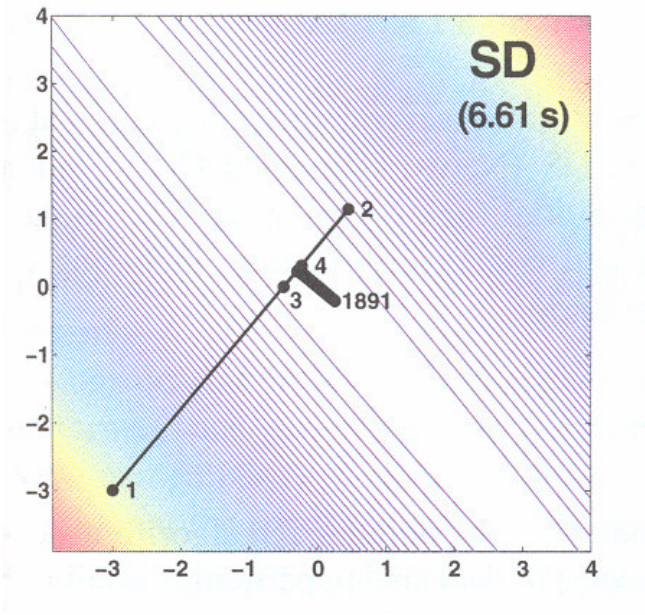
$$= -\begin{pmatrix} g_1 & g_2 & g_3 & \dots & g_n \end{pmatrix} \begin{pmatrix} g_1 \\ g_2 \\ g_3 \\ \dots \\ g_n \end{pmatrix}$$

$$= -\left[ (g_1)^2 + (g_2)^2 + (g_3)^2 + \dots + (g_n)^2 \right] < 0$$

# Steepest Descent (SD) Methode in der Praxis

- Ist die **einfachste** häufig verwendete Methode
- folgt wie oben erwähnt jeweils dem **negativen Gradienten g** (also der grössten Kraft)

$$\mathbf{d} = -\mathbf{g}$$

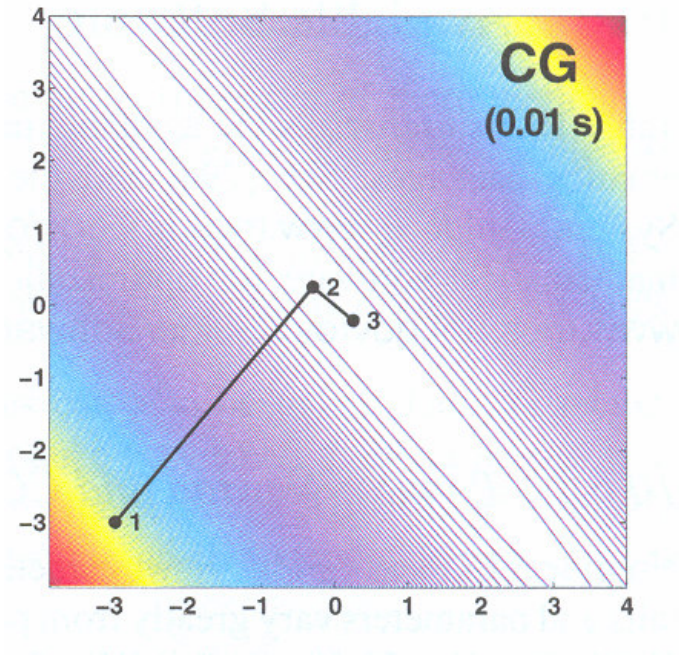



- man legt meist einen minimalen und maximalen Wert für die Verschiebung der Koordinaten fest
- jenach, wie steil der Gradient ist, wird die **Schrittweite** vergrössert oder verkleinert
- SD ist die **schnellste** Methode, wenn man eine schlechte Startgeometrie besitzt (z.B. Proteinstruktur aus .pdb Datei)
- **konvergiert langsam** in der Nähe des Minimums
- kann dort um das Minimum oszillieren

# Conjugate Gradient (CG) Methode

- verwendet die ‘Geschichte’ der Minimierung, also den vorherigen Wert $\mathbf{d}_{i-1}$.

$$\mathbf{d}_i = -\mathbf{g}_i + \beta_{i-1}\,\mathbf{d}_{i-1}$$




- Im Gegensatz zu SD wird also implizit die **Information** der **zweiten Ableitungen** verwendet um die Suche zu steuern.

- Es gibt viele Varianten von CG wie die Fletcher-Reeves und Polak-Ribiere Methoden. (siehe A. Leach “Molecular Modelling)

- CG konvergiert wesentlich schneller in der Nähe des Minimums als SD!

# Methoden mit zweiter Ableitung

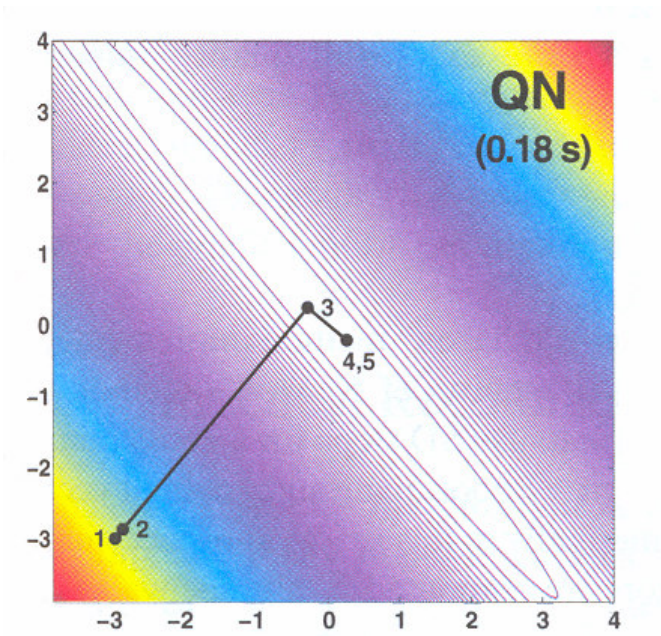
- Die zweite Ableitung der Energie *E* bezüglich der (kartesischen) Koordinaten ***r*** – die Hess'sche Matrix ***H(r)*** – bestimmt den Suchpfad.

  Entwickle *E(r)* um ***r***

$$E(\mathbf{r}) \cong E(\mathbf{r}_0) + \mathbf{g}^t(\mathbf{r}-\mathbf{r}_0) + \frac{1}{2}(\mathbf{r}-\mathbf{r}_0)^t \mathbf{H}(\mathbf{r}-\mathbf{r}_0)$$

  Dann ist der Newton-Raphson Schritt

$$\Delta\mathbf{r}' = \sum_i \Delta x_i' \qquad , \Delta x_i' = -\frac{f_i}{\varepsilon_i}$$




- $f_i$ ist die Projektion des Gradienten entlang des Hess'schen Eigenvektors mit Eigenwert $\varepsilon_i$.

- Ist rechenaufwendiger, aber gewöhnlich schnell und zuverlässig, besonders in der Nähe des Minimums.

- Varianten: Quasi-Newton (QN), Newton-Raphson, Davidon-Fletcher-Powell (DFP), Broyden-Fletcher-Goldfarb-Shanno (BGFS) und Eigenvector Following

# Trust-Region basierender Descent Algorithmus [A2]

Nimm an, daß in einer inneren Region eine quadratische Funktion $q_k$ als Näherung an $f$ existiert, der man "vertrauen kann". Es reicht dann aus, das Minimum dieser quadratischen Funktion zu finden.

`Algorithmus:`

Von einem gegebenen Anfangspunkt $\boldsymbol{x}_0$ aus, führe für $k$ = 0, 1, 2, … die folgenden Schritte so lange aus bis Konvergenz eintritt.

(1) Überprüfe $\boldsymbol{x}_k$ auf Konvergenz

(2) Berechne einen Schritt $\boldsymbol{s}_k$ durch Lösen des Subproblems $\min_s \{q_k(s)\}$

wobei $q_k$ das quadratische Model der Zielfunktion ist:

$$q_k(\mathbf{s}) = f(\mathbf{x}_k) + \mathbf{g}_k^T \mathbf{s} + \frac{1}{2} \mathbf{s}^T \mathbf{H}_k \mathbf{s}$$

Dies gilt für $\boldsymbol{s}$ innerhalb einer Schranke $\Delta_k > 0$.

Ausgedrückt mit einer Skalierungsmatrix $\boldsymbol{D}_k$ gilt $\|\mathbf{D}_k \mathbf{s}\| < \Delta_k$

(3) Setze $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k$ und $k \rightarrow k + 1$ und gehe zurück nach (1)

# praktische Empfehlungen

- Benutze **viele unterschiedliche Startpunkte** (Startgeometrien)
- Vergleiche Ergebnisse mehrerer Algorithmen (bzw. mehrerer Kraftfelder)
- überprüfe Eigenwerte im Minimum (zweite Ableitung erforderlich).

  Bis auf 6 Eigenwerte, die fast gleich Null sein sollten und die Translations- und Rotationsinvarianz wiederspiegeln, sollten alle positiv sein.

[Schlick, Kap. 10.7]

Bei der Optimierung von Proteinstrukturen ausgehend von den Koordinaten aus der zugehörigen .pdb Datei empfiehlt sich folgende Strategie:

1. 100-500 Schritte mittels steepest descent (grobe Optimierung, v.a. Torsionswinkel und nichtbindende Wechselwirkungen)
2. anschließende Optimierung mit einem conjugate gradient Verfahren (schneller in der Nähe eines Minimums)

[Leach “Molecular Modelling”, Kap.4 Energy minimization]

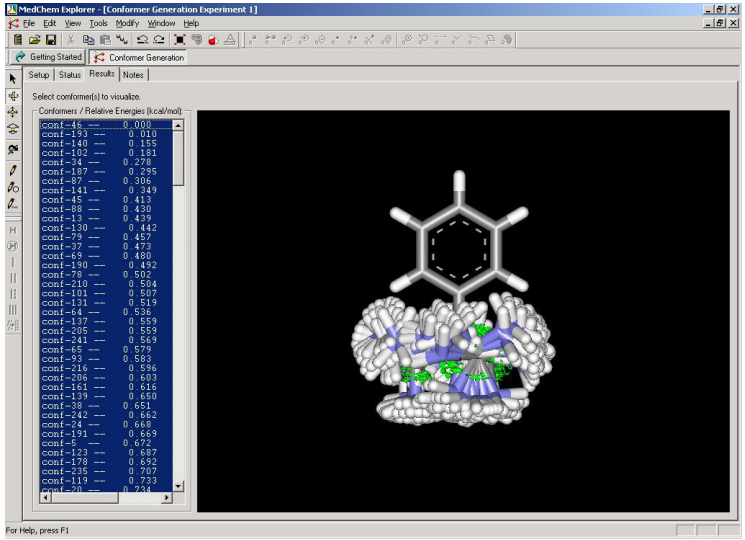
# Verfahren um das globale Energie-Minimum zu finden

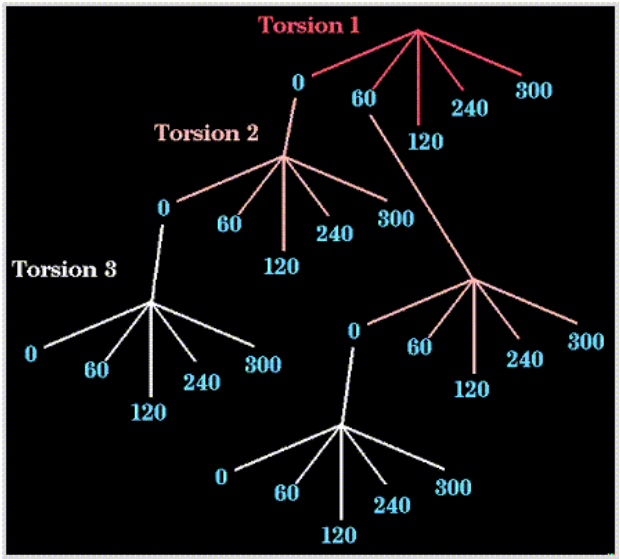
- **Systematische Variation** der Torsionswinkel
- Randomization-minimization (**Monte Carlo**)
- **Moleküldynamik** (*Newton'sche Bewegungsgleichung*)
- **Simulated Annealing** (reduziere Temperatur während MD Simulation)
- **Genetische Algorithmen** (man startet wird einer Menge von Konformationen; kleine Veränderungen; behalte die der geringsten Energie; wiederhole diese Schritte)
- Reine **Zufallssuche** (funktioniert am schlechtesten bei großen Molekülen, ineffektiv)

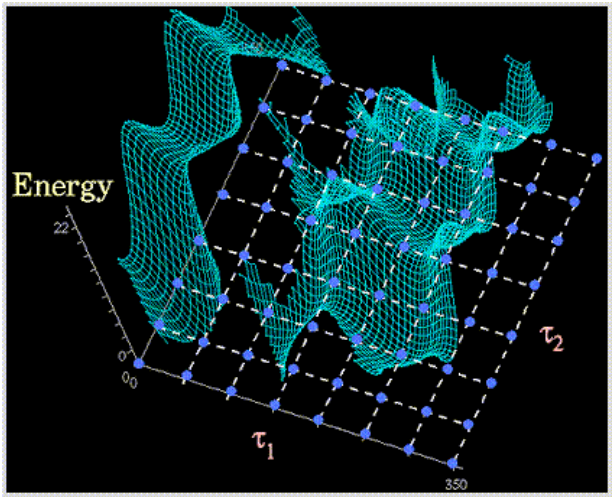
# Systematische Variation der Torsionswinkel

Für *N* rotierbare Bindungen eines Moleküles, die mit Auflösung *d* abgesucht werden sollen, gibt es $N^d$ Konformationen.

Dies geht nur bei kleiner Anzahl von Freiheitsgraden, da sonst kombinatorische Explosion.




NIH guide to molecular modelling
http://cmm.info.nih.gov/modeling/guide_documents/sybyl

# Systematische Variation der Torsionswinkel: Gridsuche

GRIDSEARCH

```
For θ1 = 120°, 240°, 360° {
    For θ2 = 120°, 240°, 360° {
        ⋮
        For θN = 120°, 240°, 360° {
            Structure S = InstantiateConformation([θ1, θ2, ..., θN]).
            Output S if it satisfies all the constraints.
        }
        ⋮
    }
}
```

Vorgabe: Struktur soll bestimmte Distanzen aus NMR-Messung erfüllen.
Bsp: Proteinstrukturbestimmung bei kleinen Proteinen

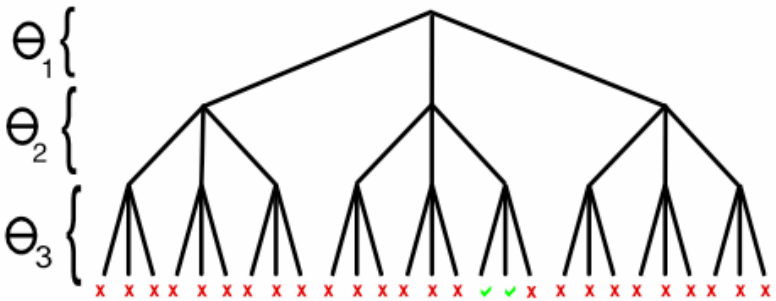
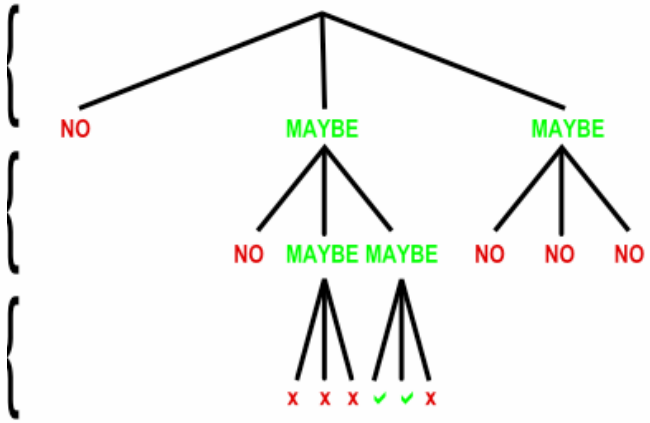



Figure 2-1: A depiction of how Gridsearch might sample three angles per bond on a molecule with three bonds. All branches are explored and evaluation only occurs after angles have been assigned to all bonds.

Lisa T. Kellogg
PhD thesis, MIT

# Systematische Variation der Torsionswinkel: Baumsuche

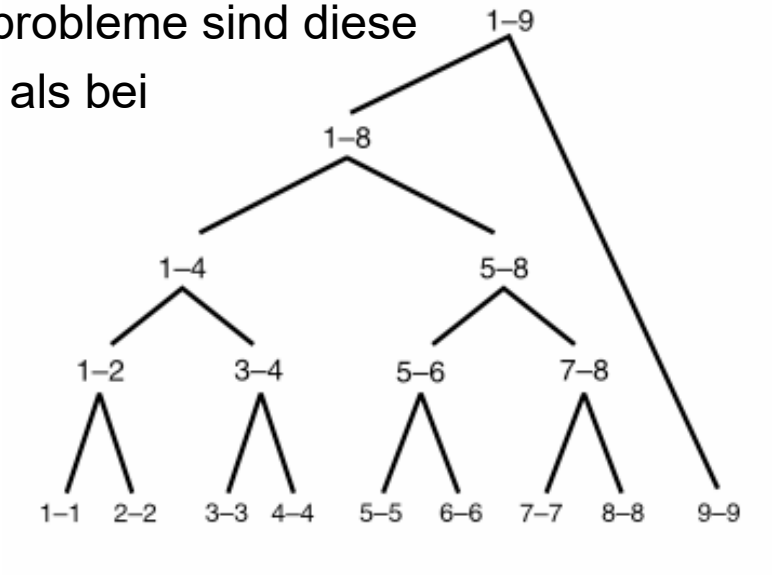
```
TREESEARCH
For θ1 = 120°, 240°, 360° {
   Structure s1 = INSTANTIATECONFORMATION([θ1]).
   If structure s1 satisfies all constraints active on it,
   Then For θ2 = 120°, 240°, 360° {
      Structure s2 = INSTANTIATECONFORMATION([θ1, θ2]).
      If structure s2 satisfies all constraints active on it,
         Then For θ3 = 120°, 240°, 360° {
               ⋮
               Then For θN = 120°, 240°, 360° {
                  Structure SN =
                     INSTANTIATECONFORMATION([θ1, θ2, θ3, ..., θN]).
                  Output S if it satisfies all the constraints.
               }
               ⋮
   } } }
```




Lisa T. Kellogg
PhD thesis, MIT

# moderne Verfahren: Teile und Herrsche (Divide-and-Conquer)

- schließe Regionen des Konformationsraums aus aufgrund der Bewertung von Unterproblemen niedriger Dimensionalität
- verbessere Baumsuche
  - bewerte jedes Stück bevor neue Aufgabe in Angriff genommen wird (bestimme die beste erzielbare Energie in diesem Zweig, verwerfe von vorneherein schlechte Zweige “dead end elimination”)
  - nachdem das Unterproblem gelöst ist, speichere dessen Ergebnis
  - durch Zerlegung in Unterprobleme sind diese im Mittel leichter zu lösen als bei der Baumsuche
- Ähnlich: A*-Algorithmus




Lisa T. Kellogg
PhD thesis, MIT

# Stochastische Methoden

Stochastische Suchverfahren, die nur wichtige Bereiche des Konformationsraums durchsuchen (**„importance sampling“**) können wesentlich effizienter für Konformationssampling in großen Molekülen sein als systematische Methoden.

(1) Beginne mit Anfangskonfiguration; minimiere diese Struktur bezüglich Energie

(2) Wähle beliebige Anzahl an Torsionswinkeln dieser Konformation und variiere sie zufällig. Dann minimiere die Konformation

(3) Benutze ein Energiekriterium um zu entscheiden, ob die neue Konformation akzeptiert wird. Falls ja, fahre fort, sonst gehe zurück zu 1.

(4) Vergleiche die neue Struktur gegen die Menge aller alten Strukturen. Falls es eine neue Konformation ist, speichere sie ab. → Clustering

(5) Gehe zurück zu Schritt 2

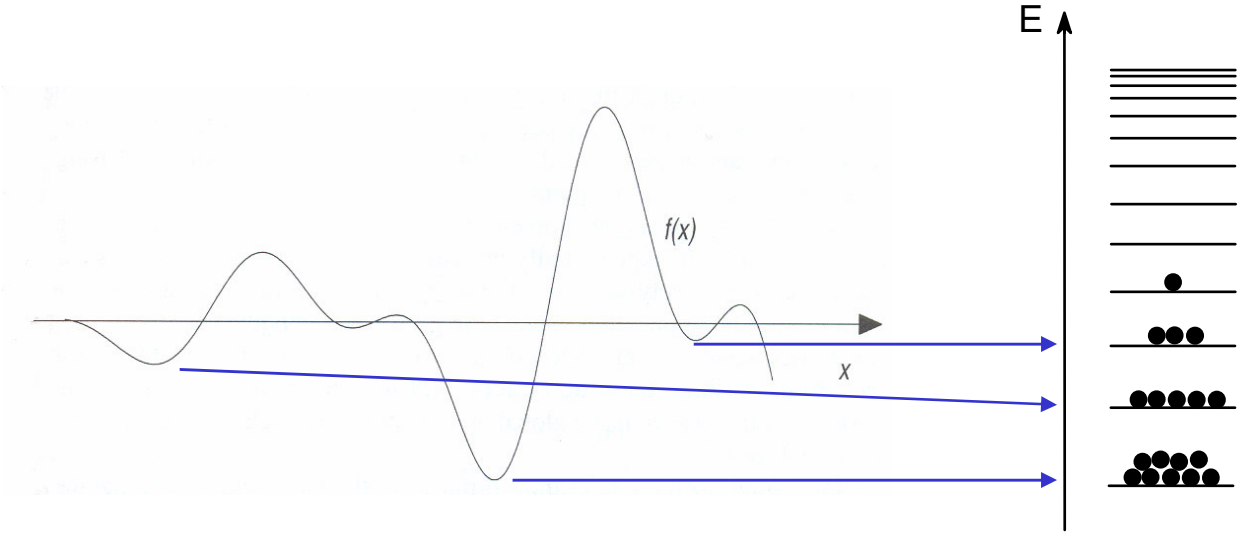
(6) Beende die Suche wenn keine neuen Strukturen mehr gefunden werden, bzw. keine neuen Cluster vorkommen.

# Anmerkungen zu Strukturen minimaler Energie

- Was bedeutet die Struktur des globalen Energieminimums eigentlich?

  Sie ist bei Raumtemperatur womöglich gar nicht populiert/besetzt/zugänglich.
- Benutzen Reaktionen/Wechselwirkungen notwendigerweise diese Geometrien minimaler Energie? (z.B. Protein-Protein-Komplexe)
- Welche anderen Konformationen niedriger Energie sind verfügbar?

  → **Boltzmann-Ensemble/Verteilung**




Die Boltzmann-Statistik/Verteilung kann immer dann eingesetzt werden, wenn die Energiezustände beliebig häufig besetzt werden können.

# Boltzmann-Verteilung

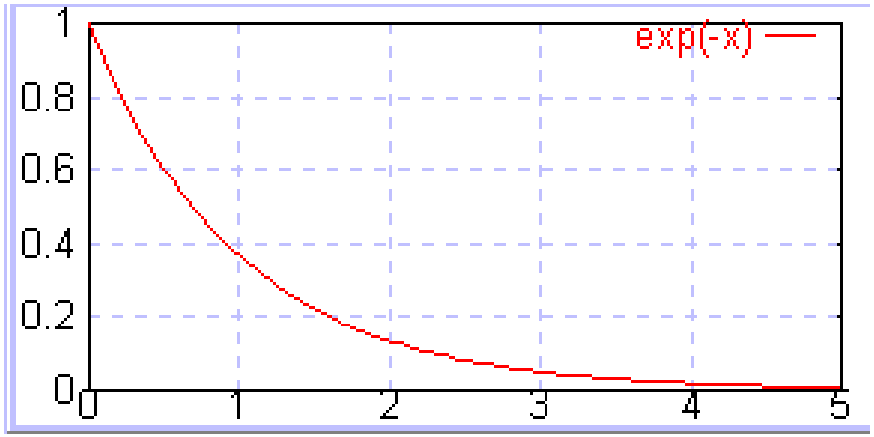
(1) In einem System mit *N* Teilchen sei **Teilchenzahl konstant**.

(2) Gesamtenergie des Systems sei konstant.

D.h. es gibt Energieaustausch zwischen den Teilchen, aber nicht mit der Umgebung.

(3) Wenn solch ein System im Gleichgewicht ist, ist die Energie der Teilchen *E* entsprechend einer Boltzmann-Verteilung populiert:

$$N(E) = N_0 \exp\left(-\frac{E}{E_0}\right)$$




Boltzmann-verteilte Systeme findet man in vielen Bereichen der Physikalischen Chemie.

# Phasenraum

Wie definiert man ein Ensemble aus *N* Teilchen im Raum und ihren jeweiligen Geschwindigkeiten sinnvoll?

Ein System aus *N* Teilchen besitzt *3N* Koordinaten und *3N* Geschwindigkeiten.

Daher besetzt es zu jedem Moment einen Punkt **X** im 6N-dimensionalen **Phasenraum**. Dessen Koordinaten sind:

$\mathbf{X} = (x_1, y_1, z_1, p_{x,1}, p_{y,1}, p_{z,1}, x_2, y_2, z_2, p_{x,2}, p_{y,2}, p_{z,2}, \ldots, x_N, y_N, z_N, p_{x,N}, p_{y,N}, p_{z,N})$

Dabei kann man separat die Orts- und Impulskoordinaten definieren als

$$\mathbf{q} = (x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_N, y_N, z_N)$$

$$\mathbf{p} = (p_{x,1}, p_{y,1}, p_{z,1}, p_{x,2}, p_{y,2}, p_{z,2}, \ldots, p_{x,N}, p_{y,N}, p_{z,N})$$

und damit diesen Punkt im Phasenraum als $\mathbf{X} = (\mathbf{q},\mathbf{p})$ darstellen.

Der Impuls(vektor) enthält sowohl Information über die Geschwindigkeit, als auch über die Masse eines Teilchens: $p = m \cdot v$

# Statistische Mechanik (I)

Die statistische Mechanik behandelt Systeme mit **vielen** (im Grunde unendlich vielen) **Freiheitsgraden**. Diese sollen durch wenige **Makrovariablen** beschrieben werden. Eine besondere Rolle spielt hierbei die Makrovariable **Entropie** (die einem Maximum zustrebt).

Die Freiheitsgrade sind für Biomoleküle speziell die Positionen **q** und Impulse **p** = m· **v** der Atome (mit deren Masse m und ihren Geschwindigkeiten **v**).

In klassischen Systemen (d.h. ohne Berücksichtigung von Quanteneffekten) kommutieren alle Freiheitsgrade miteinander. [q,p] = 0.

Den Raum, der durch die Freiheitsgrade aufgespannt wird, nennt man den **Phasenraum**. Jeder Punkt im Phasenraum bezeichnet eine bestimmte Konfiguration des Systems.

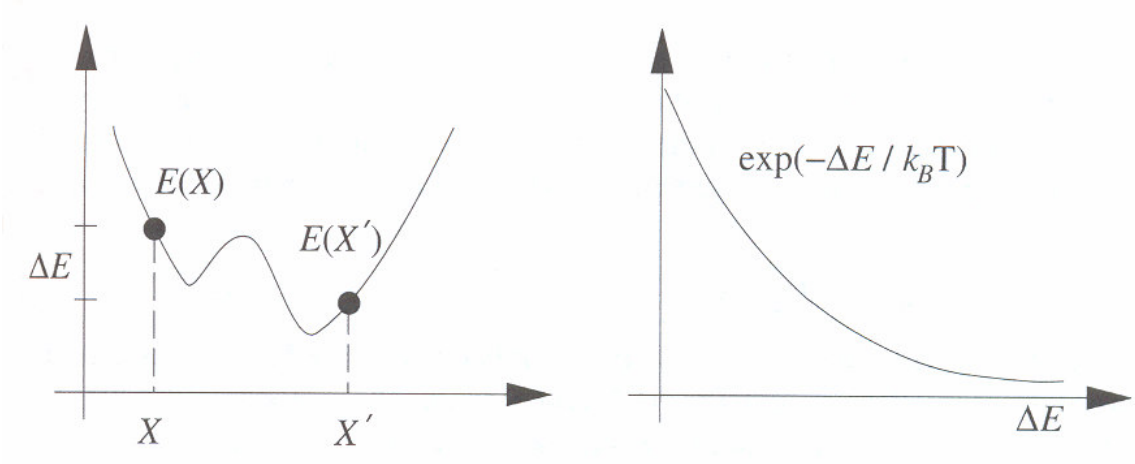
Im zeitlichen Verlauf beschreibt das System einen Pfad im Phasenraum, der durch die (Newtonschen) **Bewegungsgleichungen** bestimmt wird.

# Phasenraumdichte (I)

Die Wahrscheinlichkeitsdichte im Phasenraum (= kurz die “Phasenraumdichte”) ist im kanonischen Ensemble proportional zum Boltzmann-Faktor:

$$\rho_{NVT}(X) \propto \exp(-\beta E(X))$$

wobei *E* die Gesamtenergie des Systems ist und $\beta = k_B T$.




Für zwei Zustände des Systems X und X’ lautet das Verhältnis ihrer Wahrscheinlichkeiten:

$$\frac{\rho_{NVT}(X)}{\rho_{NVT}(X')} = \exp(-\beta \Delta E),$$

$$\Delta E = E(X) - E(X')$$

# Phasenraumdichte (II)

Der Normalisierungsfaktor der ersten Gleichung ist die Zustandssumme des gesamten Phasenraums (Raum der 3N Koordinaten und 3N Geschwindigkeiten):

$$\rho_{NVT}(X) = \frac{1}{(h^{3N})N!} \frac{\exp(-\beta E(X))}{Q_{NVT}}$$

$$Q_{NVT} = \frac{1}{(h^{3N})N!} \int \exp(-\beta E(x))dx$$

Der **Erwartungswert** einer Observablen *A* des Systems lässt sich darstellen als:

$$\langle A(x) \rangle_{NVT} = \int \rho_{NVT}(x) A(x) dx$$

Im Metropolis-Algorithmus erzeugt man eine geeignete Markov-Kette von Konfigurationen, so dass der Erwartungswert von *A* als einfacher Mittelwert folgt:

$$\langle A(x) \rangle_{NVT} = \lim_{M \to \infty} \frac{1}{M} \sum_{i=1}^{M} (A(x_i))$$

(Bei einer Markov-Kette hängt der nächst folgende Zustand nur von dem vorherigen ab)

# Markov-Kette

Betrachte die Markov-Kette von $N$ molekularen Zuständen $\{X_1, X_2, X_3, ...\}$ mit einer Verteilung $\rho_{NVT}(X)$ für $N \to \infty$.

In einer Markov-Kette gehört jeder Zustand zu einer endlichen Menge an Zuständen aus dem Zustandsraum $D_0 \in D$.

Für die konditionelle Verteilung jedes Zustands $X_{n+1}$ bezüglich aller vorherigen Zustände gilt:

$$P\{X_{n+1} \in D_0 | X_0, ..., X_n\} = P\{X_{n+1} \in D_0 | X_n\}$$

d.h. der Zustand $X_{n+1}$ hängt nur von dem unmittelbaren davorliegenden Zustand $X_n$ ab.

Der Metropolis-Algorithmus erzeugt eine stochastische und ergodische Übergangsmatrix für die Markovkette, so dass die Verteilung für jeden Zustand $X_i$ im Limit $\rho_i = \rho_{NVT}(X_i)$ ist.

So wird eine Phasenraumtrajektorie im kanonischen Ensemble (NVT konstant, Druck p und Energie E schwanken) erzeugt.

# mikroskopische Reversibilität (“detailed balance”)

Lege Übergangsmatrix fest durch Definition einer Übergangswahrscheinlichkeit $k_{ij}$ für jeden Übergang von $X_i$ nach $X_j$, so dass die **mikroskopische Umkehrbarkeit** erfüllt ist:

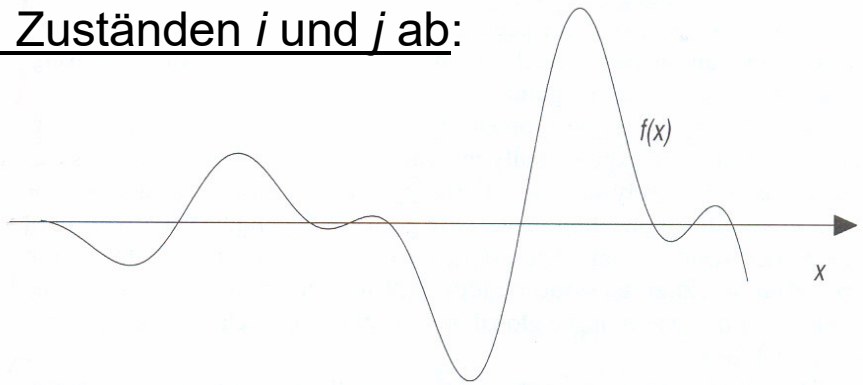
$$\rho_i k_{ij} = \rho_j k_{ji}$$

$\rho_i$ und $\rho_j$ sind die Besetzungswahrscheinlichkeiten der beiden Zustände *i* und *j*.

Im kanonischen Ensemble sind dies die Wahrscheinlichkeiten aus der Boltzmann-Verteilung. <u>Damit hängt das Verhältnis der Übergangswahrscheinlichkeiten nur vom Energieunterschied zwischen den Zuständen *i* und *j* ab</u>:

$$\frac{\rho_i}{\rho_j} = \frac{k_{ji}}{k_{ij}} \propto \exp\left(-\beta \Delta E_{ij}\right)$$

$$\Delta E_{ij} = E\left(X_i\right) - E\left(X_j\right)$$




Durch die Erfüllung dieser Bedingung ist das Gleichgewicht stationär.

Es gilt die **detailed balance**. Bei chemischen Reaktionen spricht man auch von thermodynamischer Kontrolle eines solchen Gleichgewichts.

# Metropolis Algorithmus

Die am häufigsten verwendete Technik zur Auswahl von Konformeren („importance sampling“) mittels Monte-Carlo-Methoden ist der **Metropolis Algorithmus**:

(1) konstruiere Anfangskonfiguration des Moleküls

(2) führe zufällige Änderung eines Freiheitsgrades (z.B. eines Torsionswinkel) durch.

(3) berechne Änderung der Energie $\Delta E$ aufgrund dieser Änderung der Konformation.

(4) falls $\Delta E < 0$ akzeptiere die neue Konfiguration

falls $\Delta E > 0$ berechne die Wahrscheinlichkeit $w = \exp\left(-\frac{\Delta E}{k_B T}\right)$

erzeuge Zufallszahl **r** im Intervall [0,1]

akzeptiere die neue Konfiguration, falls w ≥ **r**, sonst verwerfe sie.

Da die Boltzmann-gewichtete Energiedifferenz mit einer **Zufallszahl** verglichen wird, werden auch vereinzelt Konformere **hoher Energie** akzeptiert. Daher erhält man ein **Ensemble** (Menge) von Konformationen mit einer Energieverteilung entsprechend einer Boltzmann-Verteilung.

# Moleküldynamik-Simulation

Basiert auf **Newtonscher Bewegungsgleichung** für ein Atom *i* eines Moleküls:

$$\mathbf{F}_i = m_i \frac{\partial^2 \mathbf{r}}{\partial t^2}$$

wobei $\mathbf{F}_i$ die Kraft, $m_i$ seine Masse und $\frac{\partial^2 \mathbf{r}_i}{\partial t^2} = \mathbf{a}_i$

die Beschleunigung ist, die auf Atom *i* wirkt.

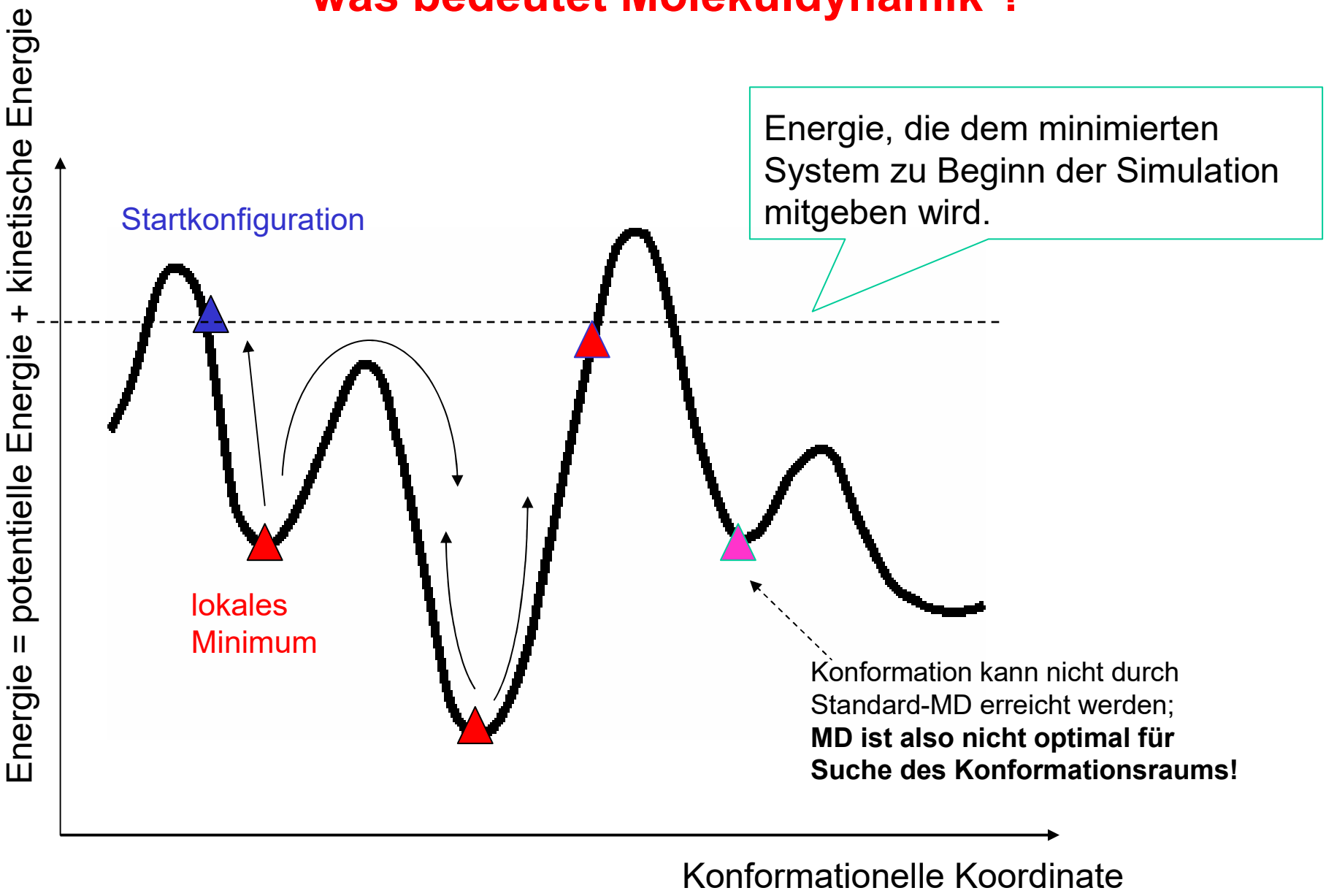
Die Kräfte lassen sich aus den Ableitungen der Energie nach den kartesischen Koordinaten ausrechnen.

• Die **Trajektorie** eines Systems ist die Aneinanderreihung der einzelnen Koordinaten und Geschwindigkeiten, also ein Film, der die Dynamik des Systems zeigt. Im Gegensatz zu Monte Carlo Verfahren bleibt die Energie erhalten.

• Ein **Ensemble** ist eine Menge von Konfigurationen, aus den man Eigenschaften des Systems berechnen kann (mittlere Energie, Wärmekapazität ...)

• mehr zu MD-Simulationen folgt in Vorlesung 5

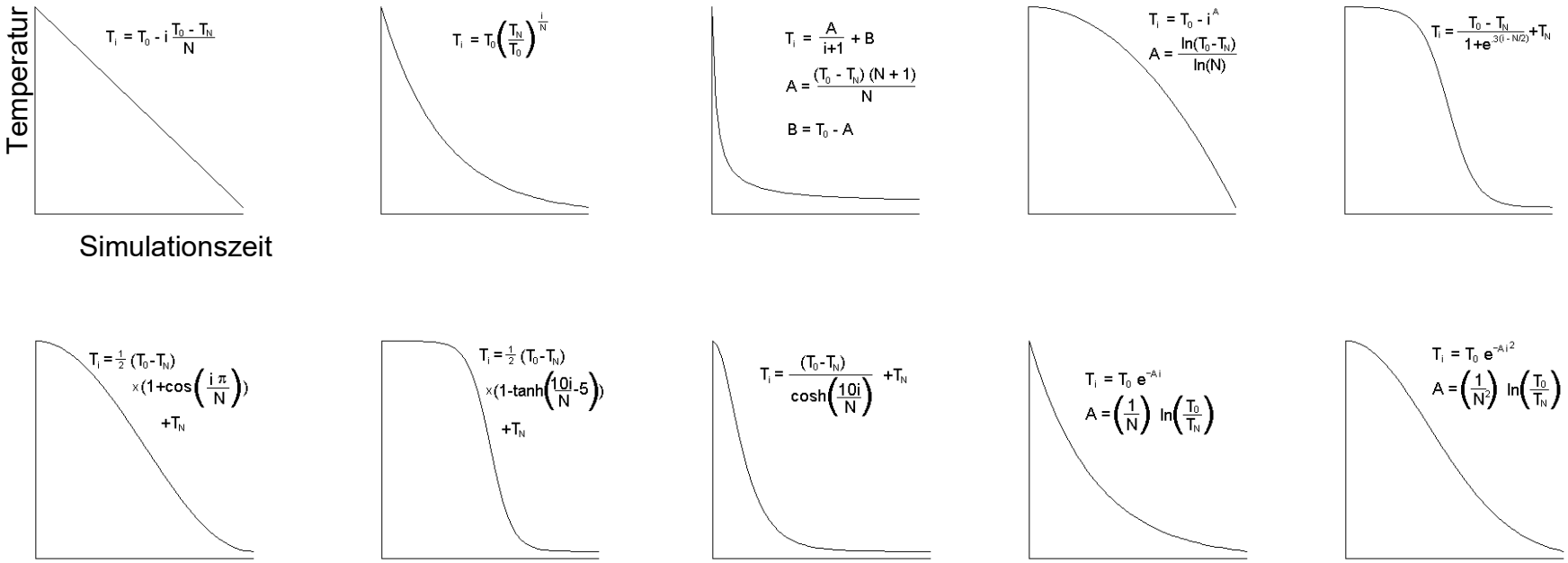
# was bedeutet Moleküldynamik ?



# Simulated Annealing

Beginne Konformationssampling (z.B. mit Moleküldynamik) bei **hoher Temperatur** um Energiebarrieren leicht zu überwinden. Kühle Simulationstemperatur dann ab.

Viele verschiedene **Abkühlstrategien** möglich:

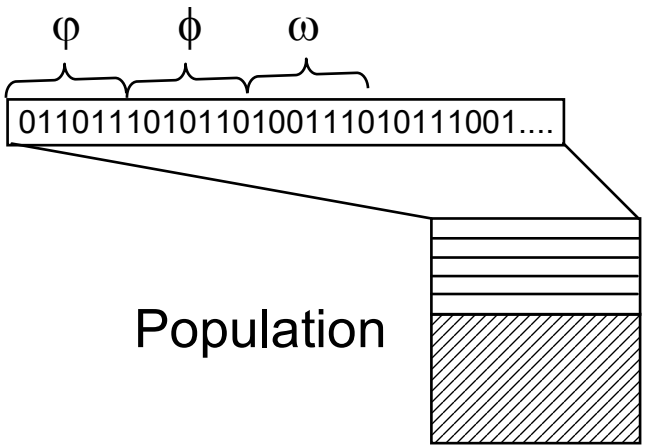



Davon ist keine “richtig” oder “falsch”. Wichtig ist, was praktisch funktioniert.

http://members.aol.com/btluke/simanf1.htm

# Genetische Algorithmen (GA) I

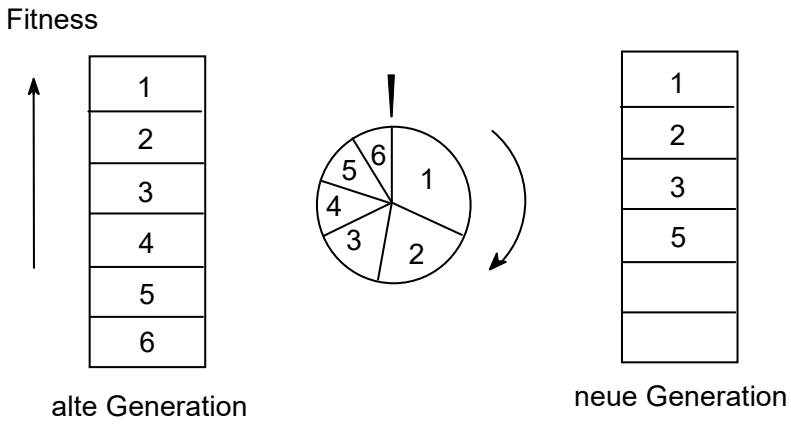
- Genetische Algorithmen basieren auf dem Prinzip der Vererbung und dem Überleben des am besten Angepassten, “survival of the fittest”.
- Die nächste Generation *i+1* unterliegt natürlicher Selektion, d.h. wir behalten die *N/Faktor* Strukturen aus Generation *i* mit den niedrigsten Energien und erzeugen im Sinne der “Evolution” neue Konformationen durch kleine “Mutationen” der Elternkonformationen, also z.B. Änderungen der Bindungswinkel und Torsionswinkel.
- Interessant werden GAs durch “Genaustausch” und Cross-over.



# Genetische Algorithmen (II)

Vererbung / Propagierung / Erzeugung der Folgegeneration:

Entsprechend der „Fitness“ jedes Individuums. Fittere Individuen haben eine höhere Chance weiterzukommen. Benötigt wird also eine Fitnessfunktion, hier die Energie.




Neue Individuen können auf verschiedene Weise erzeugt werden:

- Zufällige Mutationen
- cross-over (single, multiple)

# Genetische Algorithmen (III)

Bsp. lineares *n*-mer Peptid mit 2*n* Diederwinkeln des Rückgrats.

Angenommen, es wurden 2 Konformationen gefunden, in denen entweder die erste Hälfte des Peptids eine energetisch günstige Konformation einnimmt oder die zweite. Eine vorteilhafte “cross-over” Mutation kombiniert nun die zwei günstigen Hälften des Moleküls miteinander.

Praktische Tips zur Implementierung:
Genetische Algorithmen sind immer dann interessant, wenn die Fitness-funktion schnell zu berechnen ist, oder keine Ableitungen möglich sind.
→ *travelling salesman*, Platinenlayout, optimaler Stauraum

Anstelle der einer binären Stringdarstellung eines Individuums kann auch ein Array mit Fließkommazahlen verwendet werden.
Beim cross-over werden dann nur komplette Gene, sprich Zahlen ausgetauscht.
So vermeidet man zwangsläufig unsinnige cross-overs.
Die “Stärke” der Mutationen läßt sich ebenfalls gewichten.
Empirische Lösungen oder lokale Minima als Startwerte verwenden.

# Genetische Algorithmen (IV)

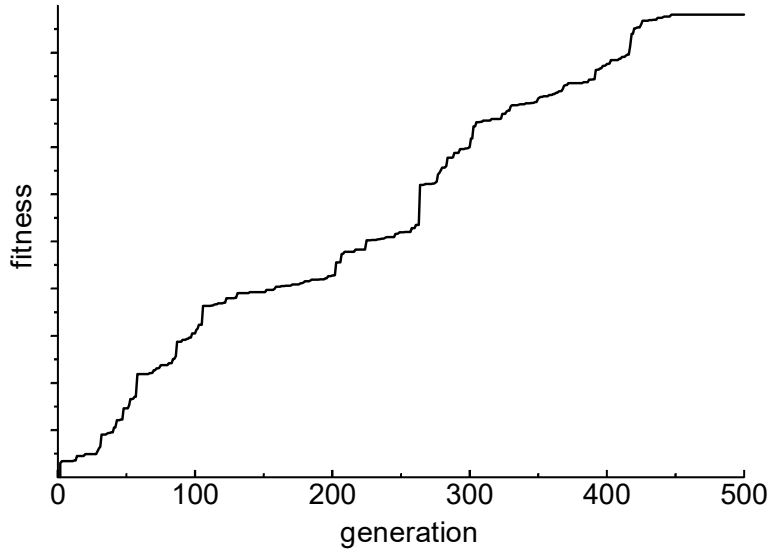
Weitere praktische Aspekte:

Die Population darf nicht zu klein sein, sonst sinkt die Diversität (Standartabweichung der Fitnessfunktion) innerhalb der Population zu schnell ab.

→ konvergiert um lokales Minimum

→ Problem wie bei Inselpopulationen

Ausweg: Stärke der Mutationen erhöhen



# Sampling des Konformationsraums

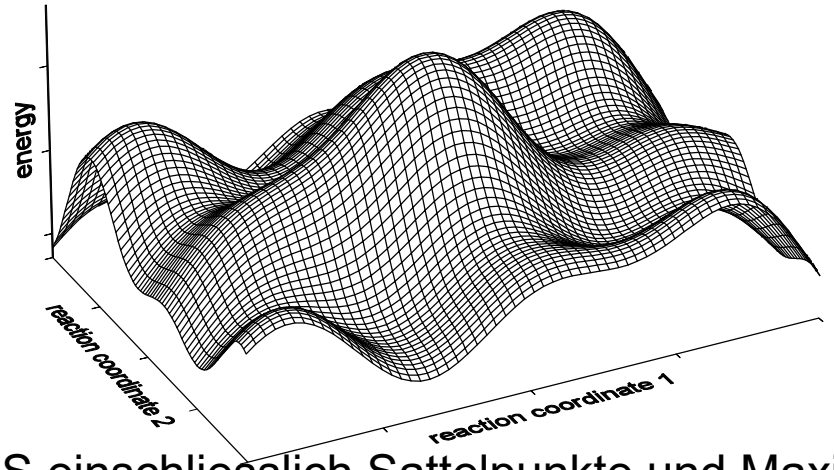
Zurück zu der anfänglich gestellten Frage:

Was möchten wir charakterisieren?

– Verfeinerung einer experimentellen Struktur bei geringer Auflösung (**lokales Energieminimum**)

– **globales Energieminimum**

– **alle Minima** geringer Energie




– die gesamte Oberfläche der PES einschliesslich Sattelpunkte und Maxima

– Einzelmoleküle oder Molekül-Ensembles?

Zu jedem dieser Problem gibt es einfache oder mächtige Methoden, die die Lösung prinzipiell finden können.

In den Fällen grosser Moleküle mit vielen Freiheitsgraden ist eine **perfekte Lösung** jedoch **oft nicht praktikabel**.

# abgeschlossene und nicht abgeschlossene Systeme

Wenn ein System thermisch und mechanisch **abgeschlossen** ist, also von seiner Umgebung isoliert, dann bleibt seine **interne Energie** zeitlich konstant.

Warum?

Wenn ein System nicht abgeschlossen ist, wird es mit der Zeit die Temperatur seiner Umgebung annehmen (die wir als konstant annehmen).

Physikalische Grössen, die entweder konstant sind oder deren Durchschnittswert durch die Umgebung bestimmt wird, bezeichnet man als **Systemparameter**.

Unterschiedliche experimentelle Umstände sorgen dafür, dass unterschiedliche Parameter konstant gehalten werden (z.B. Druck p, Volumen V, Temperatur T).
Bei biologischen Systemen sind zumeist p und T konstant.
In der Theorie der statistischen Physik entsprechen diese Fälle unterschiedlichen **Ensembles**.

# Ensembles $\leftrightarrow$ Simulationen

Ein **Ensemble** ist eine gedankliche Konstruktion: wir denken uns eine sehr große Anzahl *M* physikalisch identischer Kopien des isolierten Originalsystems, die sich in beliebigen Bewegungszuständen befinden können, dargestellt durch **Ensemblepunkte** (*q*,*p*) im Phasenraum. Jeder Ensemblepunkt charakterisiert den Bewegungszustand eines Ensemblemitglieds. Wir wollen annehmen, daß wir das Ensemble durch eine **Ensembledichte** $\rho(q,p,t)$ beschreiben können: $\rho(q,p,t)d\Gamma$ soll der Bruchteil der *M* Phasenraumpunkte sein, der sich zur Zeit *t* im Phasenraumelement $d\Gamma = dp\, dq$ bei (*q*,*p*) aufhält. Damit muss auch die Ensembledichte auf den Wert 1 normiert sein:

$$\int d\Gamma \rho(q,p,t) = 1$$

Diese Normierung erlaubt auch die Interpretation, daß $\rho(q,p,t)d\Gamma$ die *Wahrscheinlichkeit* ist, den Ensemblepunkt eines beliebig herausgegriffenen Ensemblemitgliedes im Element $d\Gamma$ bei (*q*,*p*) zu finden.

Zur Simulation von klassischen Vielteilchensystemen gibt es im Wesentlichen zwei Methoden: Monte Carlo und Moleküldynamik.
Um diese Simulationstechniken an die experimentellen Situationen anzupassen ist in jedem Fall eine sorgfältige Betrachtung notwendig, welches Ensemble dazu geeignet ist, d.h. in welchem Ensemble man simulieren muss.

# Ergoden-Hypothese.

Das grundlegende Postulat der statistischen Mechanik gilt für Systeme mit konstanter Energie $E$, Volumen $V$ und Teilchenzahl $N$. (Hier: Druck $p$ fluktuiert)

Das Postulat besagt, dass alle für das System zugänglichen Zustände (die eine bestimmte Energie, Volumen und Teilchenzahl besitzen) im Verlauf der Zeit mit gleicher Häufigkeit = Wahrscheinlichkeit eingenommen werden.
Dies ist die **Ergoden-Hypothese**.

Daher kann der zeitliche Mittelwert $\overline{A}$ einer physikalischen Messgrösse $A$ (also einer „Observablen“) durch den Mittelwert über alle zugänglichen Zustände ersetzt werden $\langle A \rangle$.

# Ensemble-Mittelwerte

$$\langle A \rangle = \frac{\sum_{\{X|E\}} A(X)}{\sum_{\{X|E\}}} = \frac{\sum_X A(X)\delta[H(X)-E]}{\sum_{\{X\}} \delta[H(X)-E]} = \overline{A}$$

Hier ist *H(X)* der **Hamiltonian**, der die Energie des Systems an einem Punkt X im Phasenraum angibt.
Damit gilt $H = T + V$, also die Summe aus den kinetischen und potentiellen Energien $T$ und $V$.

Der Nenner bewirkt die entsprechende Normierung. Die Summe $\sum_{\{X|E\}}$ bezeichnet eine Summe über alle Zustände X mit einer bestimmten Energie $E$. In der rechten Summe wird diese Einschränkung durch die Delta-Funktion erreicht. (Zusätzlich ist das System auf ein bestimmtes Volumen und auf eine bestimmte Teilchenzahl eingeschränkt.)

Der Mittelwert $\langle\ \rangle$ heisst der Ensemble-Mittelwert.

Das Ensemble *NVE* = konstant heisst das **mikrokanonische Ensemble**.

# Entropie

Die Anzahl an Zuständen Ω mit Energie *E* ist

$$\Omega(N,V,E)=\sum_X \delta[H(X)-E]$$

Die **Entropie** *S* wird mittels Ω(N, V, E) und der Boltzmann-Konstante $k_B$ definiert als:

$$S(N,V,E)=k_B \ln \Omega(N,V,E)$$

(Vergleiche: Shannon-Entropie ist ebenfalls logarithmisch von der Anzahl an Zuständen abhängig)

Die thermodynamischen Grössen Temperatur *T*, chemisches Potential $\mu$ und Druck *p* ergeben sich als Ableitungen der Entropie nach den Systemparametern:

$$T=\left(\frac{\partial S}{\partial E}\right)_{N,V}^{-1}, \quad \mu=-T\left(\frac{\partial S}{\partial N}\right)_{E,V}, \quad p=T\left(\frac{\partial S}{\partial V}\right)_{E,N}$$

Dies folgt aus dem ersten Gesetz der Thermodynamik:

$$dE=TdS-pdV+\mu dN$$

# Zuzsammenfassung V4

- Bei Energiefunktionen die sich einfach differenzieren lassen (alle Energieterme in Kraftfeldern) empfiehlt sich die Verwendung von Ableitungen bei der Energieoptimierung (Suche nach *einem* Minimum).
- Das Auffinden des globalen Minimums erfordert andere Suchstrategien. Hier sind vor allem Genetische Algorithmen zielführend.
- Sollen nur relevante Bereich des Konformationsraums gesamplet werden, dann eignen sich stochastische Methoden (z.B. Monte Carlo).
- Anstelle von Teilchenkoordinaten und deren Geschwindigkeiten kann man diese auch als Punkte im Phasenraum beschreiben.
- Alle erreichbaren Zustande konnen durch das Prinizip der mikroskopischen Reversibilitat angenommen werden.
- Der zeitliche Mittelwert einer Messgrösse entspricht deren Erwartungswert.
- Die Entropie ist hauptsachlich durch die Anzahl möglicher Zustände (insbesondere solcher mit hoher Energie) bestimmt.