Condensed Phase Simulations

Adolfo Bastida

Química Computacional y Modelización Molecular

Máster en Química Fina y Molecular
Universidad de Murcia (Spain)

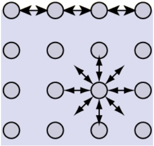
Index

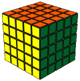
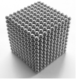
Solvents

- Most of chemical reactions take place in solution
- Good solvent
  - Inert
  - Able to solve all reactants
  - Adequate ebullition point
  - Easy to separate
- Inert $\neq$ neutral $\Rightarrow$
  - Energy fluxes
  - Intermolecular interactions
- Quantum-mechanical simulations $\Rightarrow$ computationally forbidden.
- Molecular Dynamics

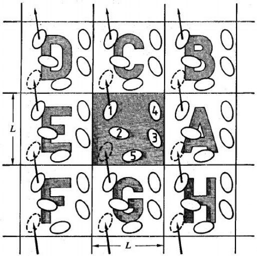
Periodic boundary conditions

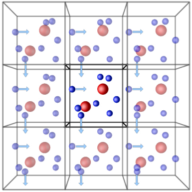
- Cubic box ⇒ Surface effects ⇒ Surface tension



-  5^3=125 molecules
  - Surface: $25+(25-5)+(25-9)+(25-9)$ $+(25-13)+(25-16)=98$ (78.4 %)
  - Bulk: $125-98=27$ (21.6 %)
-  6^3=216 molecules
  - Surface: $36+(36-6)+(36-11)+(36-11)$ $+(36-16)+(36-20)=152$ (70.4 %)
  - Bulk: $216-152=64$ (29.6 %)
-  10^3=1000 mol.
  - Surface: $100+(100-10)+(100-19)+(100-19)$ $+(100-28)+(100-36)=488$ (48.8 %)
  - Bulk: $1000-488=512$ (51.2 %)

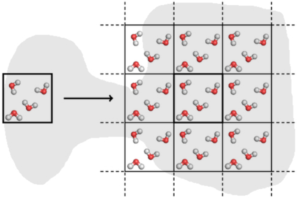
Periodic boundary conditions

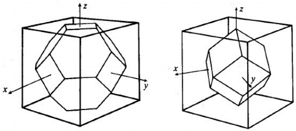




- Central box $\Rightarrow$ infinite lattice
- Periodic image in each box moves in the same way
- If a molecule leaves a box, one of its images will enter through the opposite face
- The number density in all boxes is constant
- Only $(\boldsymbol{r}_i, \boldsymbol{p}_i)$ of the molecules in the central box are stored

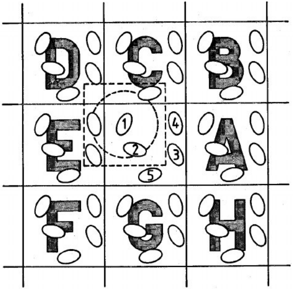
Periodic boundary conditions





- Special conditions
  - Phase transitions
  - Supercritical fluids
- Test ⇒ increase box size
- Other boxes ⇒ space filling polyhedra
  - Rombic dodecahedron
  - Truncated octahedron

Periodic boundary conditions



- Infinite number of interactions (potential energy and forces)
- Minimun Image Convention ⇒ each molecule interacts with the closest periodic image of the other $N-1$ molecules
- Define a box centered in molecule 1 ⇒ interacts with 2, 3_{E}, 4_{E}, and 5_{C}
- Pairwise-additive interactions ⇒ $\frac{1}{2}N(N-1)$ terms

Hamilton's equations

- Newton's second law

$$\boldsymbol{F}_i = m_i\, \boldsymbol{a}_i = m_i\, \frac{d^2 \boldsymbol{r}_i}{dt^2} \Rightarrow \text{Second order differential equation}$$

- Hamilton's equations

$$\left.\begin{array}{l} \dot{\boldsymbol{r}}_i = \boldsymbol{p}_i / m_i \\ \dot{\boldsymbol{p}}_i = \boldsymbol{F}_i \end{array}\right\} \Rightarrow \text{Coupled first order differential equations}$$

- Equivalent formulations $\Rightarrow \boldsymbol{F}_i = \dot{\boldsymbol{p}}_i = m_i\, \ddot{r}_i = m_i\, \boldsymbol{a}_i$

- Forces $\Rightarrow \boldsymbol{F}_i = -\nabla^2_{\boldsymbol{r}_i} V = -\left(\frac{\partial V}{\partial x_i}, \frac{\partial V}{\partial y_i}, \frac{\partial V}{\partial z_i}\right)$

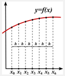
Verlet algorithm

$$\mathbf{r}(t) = \mathbf{r}(t_0) + \underbrace{\dot{\mathbf{r}}(t_0)}_{\mathbf{v}(t_0)}(t-t_0) + \frac{1}{2}\underbrace{\ddot{\mathbf{r}}(t_0)}_{\mathbf{a}(t_0)}(t-t_0)^2 + \cancel{O((t-t_0)^3}$$

$$= \mathbf{r}(t_0) + \mathbf{v}(t_0)(t-t_0) + \frac{1}{2}\mathbf{a}(t_0)(t-t_0)^2 \qquad (1)$$

$$\mathbf{r}(t\pm\delta) = \mathbf{r}(t_0) + \mathbf{v}(t_0)(t\pm\delta-t_0) + \frac{1}{2}\mathbf{a}(t_0)(t\pm\delta-t_0)^2 \qquad (2)$$

$$\begin{aligned}
(1),(2) \Rightarrow \mathbf{r}(t+\delta)+\mathbf{r}(t-\delta)-2\,\mathbf{r}(t) &= \cancel{\mathbf{r}(t_0)}+\cancel{\mathbf{v}(t_0)(t+\delta-t_0)}+\tfrac{1}{2}\mathbf{a}(t_0)(t+\delta-t_0)^2 \\
&+\cancel{\mathbf{r}(t_0)}+\cancel{\mathbf{v}(t_0)(t-\delta-t_0)}+\tfrac{1}{2}\mathbf{a}(t_0)(t-\delta-t_0)^2 \\
&-\cancel{2\,\mathbf{r}(t_0)}-\cancel{2\,\mathbf{v}(t_0)(t-t_0)}-\mathbf{a}(t_0)\,(t-t_0)^2 \\
&=\tfrac{1}{2}\,\mathbf{a}(t_0)\Big(\cancel{(t-t_0)^2}+\delta^2+\cancel{2\,(t-t_0)\,\delta}\Big) \\
&+\tfrac{1}{2}\,\mathbf{a}(t_0)\Big(\cancel{(t-t_0)^2}+\delta^2-\cancel{2\,(t-t_0)\,\delta}\Big) \\
&-\cancel{\mathbf{a}(t_0)\,(t-t_0)^2}=\mathbf{a}(t_0)\,\delta^2
\end{aligned}$$



Verlet algorithm

$$\boldsymbol{r}(t+\delta) = 2\,\boldsymbol{r}(t) - \boldsymbol{r}(t-\delta) + \boldsymbol{a}(t_0)\,\delta^2$$

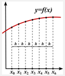
$$\boxed{\boldsymbol{r}(t_0+\delta) = 2\,\boldsymbol{r}(t_0) - \boldsymbol{r}(t_0-\delta) + \boldsymbol{a}(t_0)\,\delta^2} \Leftarrow \text{Verlet}$$

$$(1) \Rightarrow \dot{\boldsymbol{r}}(t) = \boldsymbol{v}(t) = \boldsymbol{v}(t_0) + \boldsymbol{a}(t_0)\,(t-t_0) \qquad (3)$$

$$\begin{aligned}
\boldsymbol{r}(t+\delta)-\boldsymbol{r}(t-\delta) &= \cancel{\boldsymbol{r}(t_0)}+\boldsymbol{v}(t_0)\,(\cancel{t}+\delta-\cancel{t_0})+\tfrac{1}{2}\,\boldsymbol{a}(t_0)\,(t+\delta-t_0)^2 \\
&\quad -\cancel{\boldsymbol{r}(t_0)}-\boldsymbol{v}(t_0)\,(\cancel{t}-\delta-\cancel{t_0})-\tfrac{1}{2}\,\boldsymbol{a}(t_0)\,(t-\delta-t_0)^2 \\
&= 2\,\boldsymbol{v}(t_0)\,\delta+\tfrac{1}{2}\,\boldsymbol{a}(t_0)\Big(\cancel{(t-t_0)^2}+\cancel{\delta^2}+2\,(t-t_0)\,\delta-\cancel{(t-t_0)^2}-\cancel{\delta^2}+2\,(t-t_0)\,\delta\Big) \\
&= 2\,\boldsymbol{v}(t_0)\,\delta+2\,\boldsymbol{a}(t_0)\,(t-t_0)\,\delta = 2\,\boldsymbol{v}(t)\,\delta
\end{aligned}$$

$$\boldsymbol{v}(t)=\frac{\boldsymbol{r}(t+\delta)-\boldsymbol{r}(t-\delta)}{2\,\delta}$$

$$\boxed{\boldsymbol{v}(t_0) = \frac{\boldsymbol{r}(t_0+\delta)-\boldsymbol{r}(t_0-\delta)}{2\,\delta}} \Leftarrow \text{Verlet}$$



Leap-grog algorithm

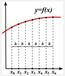
$$\text{(3) with } t \to t_0+\tfrac{\delta}{2} \Rightarrow \boldsymbol{v}(t_0+\tfrac{\delta}{2}) = \boldsymbol{v}(t_0)+\boldsymbol{a}(t_0)\,(t_0+\tfrac{\delta}{2}-t_0) = \boldsymbol{v}(t_0)+\tfrac{1}{2}\,\boldsymbol{a}(t_0)\,\delta \qquad (4)$$

$$\begin{aligned} \text{(2) with } t \to t_0 \Rightarrow \boldsymbol{r}(t_0+\delta) &= \boldsymbol{r}(t_0)+\boldsymbol{v}(t_0)\,\delta+\tfrac{1}{2}\,\boldsymbol{a}(t_0)\,\delta^2 \\ &= \boldsymbol{r}(t_0)+\delta\left(\boldsymbol{v}(t_0)+\tfrac{1}{2}\,\boldsymbol{a}(t_0)\,\delta\right) \\ &\underset{(4)}{=} \boldsymbol{r}(t_0)+\boldsymbol{v}(t_0+\tfrac{\delta}{2})\,\delta \end{aligned}$$

$$\boxed{\boldsymbol{r}(t_0+\delta) = \boldsymbol{r}(t_0)+\boldsymbol{v}(t_0+\tfrac{\delta}{2})\,\delta} \Leftarrow \text{Leap-frog}$$

$$\left.\begin{aligned} \text{(4)} \Rightarrow \boldsymbol{v}(t_0+\tfrac{\delta}{2}) &= \boldsymbol{v}(t_0)+\tfrac{1}{2}\,\boldsymbol{a}(t_0)\,\delta \\ \text{(4) with } \delta \to -\delta \Rightarrow \boldsymbol{v}(t_0-\tfrac{\delta}{2}) &= \boldsymbol{v}(t_0)-\tfrac{1}{2}\,\boldsymbol{a}(t_0)\,\delta \end{aligned}\right\} \boldsymbol{v}(t_0+\tfrac{\delta}{2})-\boldsymbol{v}(t_0-\tfrac{\delta}{2}) = \boldsymbol{a}(t_0)\,\delta$$

$$\boxed{\boldsymbol{v}(t_0+\tfrac{\delta}{2}) = \boldsymbol{v}(t_0-\tfrac{\delta}{2})+\boldsymbol{a}(t_0)\,\delta} \Leftarrow \text{Leap-frog}$$

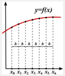


Other algorithms

$$\boxed{\begin{aligned} \boldsymbol{r}(t_0+\delta) &= \boldsymbol{r}(t_0)+\boldsymbol{v}(t_0)\,\delta+\tfrac{1}{2}\,\boldsymbol{a}(t_0)\,\delta^2 \\ \boldsymbol{v}(t_0+\tfrac{\delta}{2}) &= \boldsymbol{v}(t_0)+\tfrac{1}{2}\boldsymbol{a}(t_0)\,\delta \\ \boldsymbol{v}(t_0+\delta) &= \boldsymbol{v}(t_0+\tfrac{\delta}{2})+\tfrac{1}{2}\boldsymbol{a}(t_0+\delta)\,\delta \end{aligned}} \Leftarrow \text{Velocity Verlet}$$

$$\boxed{\begin{aligned} \boldsymbol{r}(t_0+\delta) &= \boldsymbol{r}(t)+\boldsymbol{v}(t_0)\,\delta+\tfrac{2}{3}\,\boldsymbol{a}(t_0)\,\delta^2-\tfrac{1}{6}\,\boldsymbol{a}(t_0-\delta)\,\delta^2 \\ \boldsymbol{v}(t_0+\delta) &= \boldsymbol{v}(t_0)+\tfrac{1}{3}\,\boldsymbol{a}(t_0+\delta)\,\delta+\tfrac{5}{6}\,\boldsymbol{a}(t_0)\,\delta-\tfrac{1}{6}\,\boldsymbol{a}(t_0-\delta)\,\delta \end{aligned}} \Leftarrow \text{Beeman}$$

Verlet, Leap-frog, velocity Verlet, Beeman,... ⇒ Similar global errors

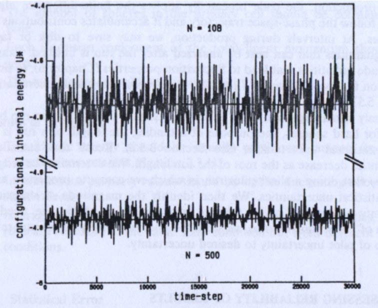


References

- L. Verlet. *Computer “experiments” on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules.* Phys. Rev. **159**, 98-103 (1967).
- W.C. Swope, H.C. Andersen, P.H. Berens and K.R. Wilson. *Computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters*. J. Chem. Phys. **76**, 637-649 (1982).
- W.H. Press, S.A: Teukolsly, W.T. Vetterling and B.P. Flanney. *Numerical recipes in Fortran 77*. Cambridge University Press. 1992. Chapter 16.7.

NVE ensemble

- *NVE* ensemble $\Rightarrow \begin{cases} N/V \;\rightarrow\; \text{density} \\ E \text{ constant} \;\rightarrow\; \text{Hamilton's equations} \end{cases}$



$$E = \frac{p^2}{2\,m} + V(\boldsymbol{r})$$

$$\dot{E} = \frac{\cancel{2}\,\boldsymbol{p}\cdot\dot{\boldsymbol{p}}}{\cancel{2}\,m} + \dot{V}(\boldsymbol{r}) = \frac{\boldsymbol{p}\cdot\dot{\boldsymbol{p}}}{m} + \nabla_{\boldsymbol{r}} V \cdot \dot{\boldsymbol{r}}$$

$$= \frac{\boldsymbol{p}}{m}\cdot\left(\dot{\boldsymbol{p}} + \nabla_{\boldsymbol{r}} V\right) = 0 \;\Rightarrow\; \dot{\boldsymbol{p}} = -\nabla_{\boldsymbol{r}} V$$

NVT ensemble

- Equilibrium ⇒ degree of freedoom ⇒ $\frac{1}{2}\,k_{\mathrm{B}}\,T$

- Temperature ⇒ $K = \sum_{i=1}^{N} \frac{1}{2}\, m_i\, \boldsymbol{v}_i^2 = 3N\,\left(\frac{1}{2}\,k_{\mathrm{B}}\,T\right) \Rightarrow T = \frac{2\,K}{3N\,k_{\mathrm{B}}}$

- Scaling

$$\left.\begin{array}{r} K = \sum_{i=1}^{N} \frac{1}{2}\, m_i\, \boldsymbol{v}_i^2 = 3N\,\left(\frac{1}{2}\,k_{\mathrm{B}}\,T\right) \\ \sum_{i=1}^{N} \frac{1}{2}\, m_i\, \boldsymbol{v}_i'^2 = 3N\,\left(\frac{1}{2}\,k_{\mathrm{B}}\,T_f\right) \end{array}\right\} \frac{\sum_i m_i\,\boldsymbol{v}_i^2}{\sum_i m_i\,\boldsymbol{v}_i'^2} = \frac{T}{T_f} \rightarrow \sum_i m_i\,\boldsymbol{v}_i^2 = \sum_i m_i\,\boldsymbol{v}_i'^2\,\frac{T}{T_f}$$

$$\boldsymbol{v}_i^2 = \boldsymbol{v}_i'^2\,\frac{T}{T_f}$$

$$\boldsymbol{v}_i' = \frac{T_f}{T}\,\boldsymbol{v}_i$$

NVT ensemble

- Coupling to a external bath [Berendsen 1984]

$$\boldsymbol{v}_i' = \lambda\, \boldsymbol{v}_i \;\rightarrow\; \lambda = \sqrt{1 + \frac{\Delta t}{\tau}\left(\frac{T_f}{T} - 1\right)}$$

- Other thermostats (Nose-Hoover,. . .)
- Equilibration (*NVT*) - Production (*NVE*)
- Other emsembles (*NPT*,. . .)

Conservation principles

- Mass ⇒ $M = \sum_i m_i$
- Energy ⇒ $E = \sum_i \frac{\boldsymbol{p}_i^2}{2\, m_i} + V(\boldsymbol{r}_i)$
- Linear momentun ⇒ $\sum_i \boldsymbol{p}_i = \boldsymbol{0}$
- Angular momentun ⇒ $\sum_i \boldsymbol{r}_i \times \boldsymbol{p}_i = \boldsymbol{0}$

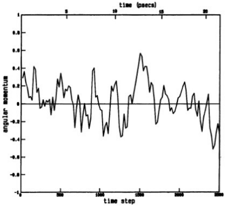


FIGURE 2.15 Fluctuations in the y-component of the angular momentum $\mathscr{Q}_y(t) = \Sigma_i^N(z_i\dot{x}_i - \dot{z}_i x_i)$, computed during a molecular dynamics simulation of 108 Lennard-Jones atoms at $\rho\sigma^3 = 0.7$, $kT/\varepsilon = 1$. Units are σ for position and $(m/\varepsilon)^{1/2}$ for velocity. Values are plotted at intervals of 20 integration time-steps; using potential parameters for argon, 20 time-steps = 0.172 psec.

Maxwell-Boltzmann distribution

- Velocity distribution

$$\rho(v_x) = \sqrt{\frac{m}{2\pi k_\mathrm{B} T}}\, e^{-\frac{1}{2} m v_x^2 / k_\mathrm{B} T} \Leftarrow \int_{-\infty}^{\infty} \rho(v_x)\, dv_x = 1$$

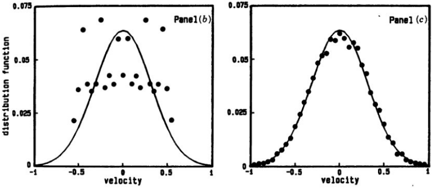


FIGURE 3.9 Approach to equilibrium for the 256-hard-sphere simulation described in Figure 3.7. (*a*) Same as Figure 3.8(*a*). (*b*) The velocity distribution $f(v_x)\,dv_x$ averaged over the first 5000 collisions of the run (points) compared to the Maxwell distribution (line) at the same kinetic energy. (*c*) The velocity distribution $f(v_x)\,dv_x$ averaged over collisions 6000–11,000 of the run (points) compared to the Maxwell distribution (line).

Static properties

- Time average

$$\langle A \rangle = \lim_{\tau \to \infty} \frac{1}{\tau} \int_0^{\tau} A(t)\, dt \approx \frac{1}{M} \sum_{k=1}^{M} A(t_k)$$

- Fluctuations

$$\delta A = A - \langle A \rangle$$

$$\begin{aligned} \langle (\delta A)^2 \rangle &= \langle (A - \langle A \rangle)^2 \rangle \\ &= \langle A^2 - 2\, A \langle A \rangle + \langle A \rangle^2 \rangle \\ &= \langle A^2 \rangle - \langle A \rangle^2 \end{aligned}$$

Static properties

- Pressure

$$P = \rho k_{\mathrm{B}} T + \frac{\langle W \rangle}{V}$$

$$W = \frac{1}{3} \sum_{i}^{N} \boldsymbol{r}_i \cdot \boldsymbol{F}_i \rightarrow \text{internal virial}$$

- Heat capacity $\Rightarrow C_V = \left(\frac{\partial E}{\partial T}\right)_V \rightarrow \; C_V^* = \left(1 - \frac{2}{3NT^{*2}} \langle (\delta K^*)^2 \rangle\right)^{-1}$
- Adiabatic compressibility $\Rightarrow \kappa = -\frac{1}{V}\left(\frac{\partial V}{\partial P}\right)_S$

$$\kappa^* = \left(7P^* - \frac{16}{3}\rho^* T^* - 8\rho^* U^* - \frac{N}{\rho^*} T^* \langle (\delta P^*)^2 \rangle\right)^{-1}$$

- Thermal pressure coefficient $\Rightarrow \gamma = \left(\frac{\partial P}{\partial T}\right)_V$

$$\gamma^* = \frac{2}{3} C_V^* \left(\rho * - \frac{1}{T^{*2}} \langle \delta K^* \delta P^* \rangle\right)$$

Static properties

- Radial distribution function $g(r)$

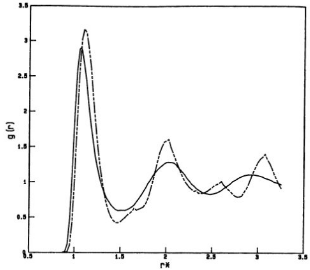


FIGURE 6.22 The radial distribution function $g(r)$ may often be used to distinguish a fluid state (continuous line) from a metastable solidlike state (broken line). These curves are from simulations that used the Lennard-Jones potential with 256 atoms at a density of $\rho\sigma^3 = 0.9$. The fluid state was at $kT/\varepsilon = 1.087$, while the metastable state was at $kT/\varepsilon = 0.80$. See Figure 5.7 for the location of these states on the Lennard-Jones phase diagram.

$$g(r) = \frac{\rho(r)}{\rho}$$

$$r \in (R, R+\delta) \rightarrow V = \frac{4}{3}\pi\left((R+\delta)^3 - R^3\right)$$

$$\rho(R) = \frac{n(R)}{V}$$

Dynamic properties

- Time correlation functions

$$\left\langle A(t_0)\, B(t_0+t)\right\rangle = \left\langle A(0)\, B(t)\right\rangle = \lim_{\tau\to\infty} \frac{1}{\tau} \int_0^{\tau} A(t_0)\, B(t_0+t)\, dt_0$$
$$\simeq \frac{1}{M} \sum_{k=1}^{M} A(t_k)\, B(t_k+t)$$

- Limits $\left\langle A(0)\, B(t)\right\rangle$

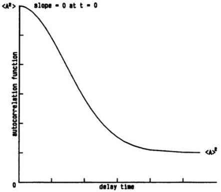


FIGURE 7.2 An autocorrelation function $\langle A(0)A(t)\rangle$ is initially equal to $\langle A^2\rangle$, but as the delay time grows, it decays, initially with zero slope, to $\langle A\rangle^2$.

$$t=0 \Rightarrow \left\langle A\, B\right\rangle \simeq \frac{1}{M} \sum_{k=1}^{M} A(t_k)\, B(t_k)$$
$$t\to\infty \Rightarrow \left\langle A\right\rangle \left\langle B\right\rangle \text{ (uncorrelated)}$$

Dynamic properties

- Transport coefficients

$$\text{Flux} = -\text{coefficient} \times \text{gradient}$$

$$\text{Fick's law} \Rightarrow N\dot{x} = -D\,\frac{\partial N}{\partial x}\begin{cases} N(x,t) \rightarrow \text{number atoms/unit volumen} \\ \dot{x}(x,t) \rightarrow \text{local velocity} \end{cases}$$

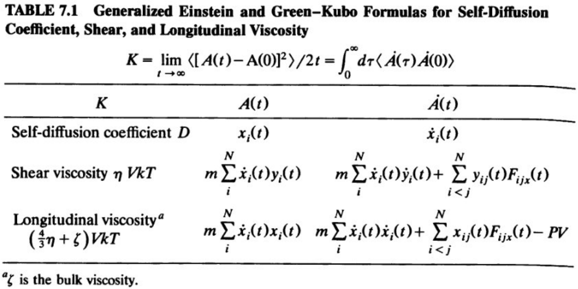


TABLE 7.1 Generalized Einstein and Green–Kubo Formulas for Self-Diffusion Coefficient, Shear, and Longitudinal Viscosity

$$K = \lim_{t\to\infty} \langle [A(t) - \mathrm{A}(0)]^2 \rangle / 2t = \int_0^\infty d\tau \langle \dot{A}(\tau)\dot{A}(0)\rangle$$

K	$A(t)$	$\dot{A}(t)$
Self-diffusion coefficient D	$x_i(t)$	$\dot{x}_i(t)$
Shear viscosity $\eta\, VkT$	$m\sum_i^N \dot{x}_i(t)y_i(t)$	$m\sum_i^N \dot{x}_i(t)\dot{y}_i(t) + \sum_{i<j}^N y_{ij}(t)F_{ijx}(t)$
Longitudinal viscosity^a $(\frac{4}{3}\eta + \zeta)VkT$	$m\sum_i^N \dot{x}_i(t)x_i(t)$	$m\sum_i^N \dot{x}_i(t)\dot{x}_i(t) + \sum_{i<j}^N x_{ij}(t)F_{ijx}(t) - PV$

$^a\zeta$ is the bulk viscosity.

Statistical Thermodynamic

- Statistical averages

$$\langle A \rangle = \frac{\int A\, e^{-\beta\, V} d\boldsymbol{r}^N}{Z} \quad \begin{cases} \beta = \frac{1}{k_\mathrm{B}\, T} \\ Z = \int e^{-\beta\, V} d\boldsymbol{r}^N \rightarrow \text{partition function} \end{cases}$$

$$= \int A \underbrace{\left(\frac{e^{-\beta\, V}}{\int e^{-\beta\, V} d\boldsymbol{r}^N} \right)}_{\substack{N(\boldsymbol{r}^N) \\ \text{probability density}}} d\boldsymbol{r}^N$$

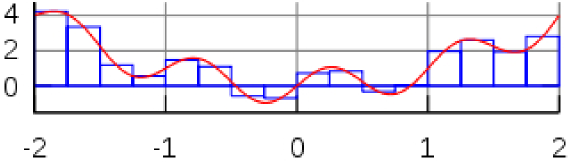
$$= \int A\, N(\boldsymbol{r}^N)\, d\boldsymbol{r}^N$$

Statistical Thermodynamic

$$\int_a^b f(x)\,dx \simeq \sum_{i=1}^{M} f\left(\frac{x_i + x_i}{2}\right) \underbrace{(x_{i+1} - x_i)}_{\delta}$$

$$= \delta \sum_{i=1}^{M} f\left(\frac{x_i + x_i}{2}\right)$$

$$= \delta \cdot M \left\langle f \right\rangle = (b-a) \left\langle f \right\rangle$$



Statistical Thermodynamic

- d dimensions $\rightarrow M^d$ grid $\left\{ \begin{array}{l} N = 10 \;\rightarrow\; d = 30 \\ M = 100 \text{ grid points} \end{array} \right\}$ $100^{30} = 10^{60}$ grid points
- Most of the grid points $\Rightarrow V \gg k_{\mathrm{B}} T \Rightarrow N(\boldsymbol{r}^N) \simeq 0$
- How to generate points in configuration space with relative probabilities proportional to the Boltzmann factor?

  → o (old) configuration with Boltzmann factor $e^{-\beta\, V(o)}$

  → small random displacement

  → n (new) configuration with Boltzmann factor $e^{-\beta\, V(n)}$

  → Accept or reject n?

The Metropolis method

- Detailed balance ⇒ Equilibrium

$$N(o)\,\pi(o\rightarrow n) = N(n)\,\pi(n\rightarrow o)$$

$$\frac{\pi(o\rightarrow n)}{\pi(n\rightarrow o)} = \frac{N(n)}{N(o)} = \frac{e^{-\beta\,V(n)}/Z}{e^{-\beta\,V(o)}/Z} = e^{-\beta\,(V(n)-V(o))} \qquad (5)$$

- The Metropolis method [Metropolis *et al* (1953)]

$$\pi(o\rightarrow n) = \begin{cases} e^{-\beta\,(V(n)-V(o))} & V(n) > V(o) \\ 1 & V(n) \leq V(o) \end{cases}$$

$$\begin{array}{ccc} V(n) > V(o) & & V(n) \leq V(o) \\ \Downarrow & & \Downarrow \\ \left\{ \begin{array}{l} \pi(o\rightarrow n) = e^{-\beta\,(V(n)-V(o))} \\ \pi(n\rightarrow o) = 1 \end{array} \right\} & \rightarrow (5) \leftarrow & \left\{ \begin{array}{l} \pi(o\rightarrow n) = 1 \\ \pi(n\rightarrow o) = e^{-\beta\,(V(o)-V(n))} \end{array} \right\} \end{array}$$

The Metropolis method

1. Select a particle at random, and calculate its energy $\mathcal{U}(\mathbf{r}^N)$.
2. Give the particle a random displacement, $\mathbf{r}' = \mathbf{r} + \Delta$, and calculate its new energy $\mathcal{U}(\mathbf{r}'^N)$.
3. Accept the move from $\mathbf{r}^N$ to $\mathbf{r}'^N$ with probability

$$\operatorname{acc}(o \to n) = \min\left(1, \exp\{-\beta[\mathcal{U}(\mathbf{r}'^N) - \mathcal{U}(\mathbf{r}^N)]\}\right). \qquad (3.2.1)$$

Algorithm 1 (Basic Metropolis Algorithm)

```
PROGRAM mc                          basic Metropolis algorithm

do icycl=1,ncycl                    perform ncycl MC cycles
  call mcmove                       displace a particle
  if (mod(icycl,nsamp).eq.0)
+    call sample                    sample averages
enddo
end
```

Comments to this algorithm:

1. *Subroutine* `mcmove` *attempts to displace a randomly selected particle (see Algorithm 2).*
2. *Subroutine* `sample` *samples quantities every* `nsamp`*th cycle.*

Algorithm 2 (Attempt to Displace a Particle)

```
SUBROUTINE mcmove                   attempts to displace a particle

o=int(ranf()*npart)+1               select a particle at random
call ener(x(o),eno)                 energy old configuration
xn=x(o)+(ranf()-0.5)*delx           give particle random displacement
call ener(xn,enn)                   energy new configuration
if (ranf().lt.exp(-beta             acceptance rule (3.2.1)
+    *(enn-eno)) x(o)=xn            accepted: replace x(o) by xn
return
end
```

Comments to this algorithm:

1. *Subroutine* `ener` *calculates the energy of a particle at the given position.*
2. *Note that, if a configuration is rejected, the old configuration is retained.*
3. *The* `ranf()` *is a random number uniform in* $[0, 1]$.

The Metropolis method

- N. Metrópilis, , A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, *Equation of State Calculations by Fast Computing Machines.* J. Chem. Phys. **21**, 1087 (1953).