MOLECULAR DYNAMICS SIMULATIONS METHODS AND APPLICATIONS

Herman J.C. Berendsen
Biophysical Chemistry, University of Groningen, the Netherlands
Groningen Institute for Biosciences and Biotechnology (GBB)

CSC, Espoo

Lecture 2, Tuesday 3 Feb, 2004

Classical dynamics allows the study of systems of 100,000 particles over time spans in the 100 ns range. This includes much (but not all) of the biologically relevant motion of proteins and of lipids in membranes, even including spontaneous aggregation. But can we distinguish the functionally relevant motions among the messy random fluctuations? Do simulations really help to understand and even predict function?

In this lecture we first look at the basic equations and algorithms to solve the motion of large molecules with atomic detail.

CLASSICAL MECHANICS

Classical mechanics is based on *Hamilton's principle of least action*: Define generalized coordinates $q_1, \ldots, q_n$. Let us have a function $L(q, \dot{q}, t)$ (the *Lagrangian*) and define the **action** S as

$$S = \int_{t_1}^{t_2} L(q, \dot{q}, t)\, dt$$

This is a *path integral*; the outcome depends on the path between q_1 and q_2.

The principle of least action says that a mechanical system follows the path of least action, given the initial and final points.

In quantum mechanics Feynman has formulated a path integral formalism that is equivalent to Schrödinger's equation:

$$\Psi(q_2, t_2) = \int G(q_2, t_2; q_1, t_1)\, \Psi(q_1, t_1)\, dq_1$$

with

$$G(q_2, t_2; q_1, t_1) = \sum_{\text{all possible paths}} \exp(iS/\hbar)$$

If $S/\hbar$ is large, the phases of different paths cancel, unless for paths close to the path with minimum S. Classically, only the path with minimum S survives.

LAGRANGIAN MECHANICS

From the principle of least action follows that

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = 0$$

Define the *generalized momentum* p_i, *conjugate to* q_i:

$$p_i = \frac{\partial L(q, \dot{q})}{\partial \dot{q}_i}$$

The *Lagrange equation of motion* follows:

$$\dot{p}_i = \frac{\partial L}{\partial q_i}$$

For *conservative force fields* the Lagrangian is

$$L = K(q, \dot{q}) - V(q)$$

Equations of motion for cartesian coordinates:

$$q_i = x_i; \quad \dot{q}_i = v_i; \quad K = \sum_{i=1}^{3N} \frac{1}{2} m_i v_i^2$$

Momentum:

$$p_i = \frac{\partial L}{\partial v_i} = m_i v_i$$

and

$$\frac{dp_i}{dt} + \frac{\partial V}{\partial q_i} = 0$$

HAMILTON FORMALISM

Define *Hamiltonian*

$$H = \sum_i p_i \dot{q}_i - L(q, \dot{q})$$

dH turns out to be a *total differential* in dp and dq:

$$dH = \sum_i p_i \, d\dot{q}_i + \sum_i \dot{q}_i \, dp_i - \sum_i \frac{\partial L}{\partial q_i} dq_i - \sum_i \frac{\partial L}{\partial \dot{q}_i} d\dot{q}_i$$

$$dH = \sum_i \dot{q}_i \, dp_i - \sum_i \dot{p}_i \, dq_i$$

Hence H is a function of p and q, and

$$\frac{\partial H(p,q)}{\partial p_i} = \dot{q}_i \quad \frac{\partial H(p,q)}{\partial q_i} = -\dot{p}_i$$

These are Hamilton's equation of motion. They use the generalized momentum, defined by the Lagrangian.

In cartesian coordinates

$$H = K + V = \sum_i p_i^2 / 2m_i + V(x)$$

$$\frac{dx_i}{dt} = \frac{\partial H}{\partial p_i} = p_i / m_i = v_i$$

$$\frac{dp_i}{dt} = -\frac{\partial V}{\partial x_i} = \text{force on i} - \text{th particle}$$

ALGORITHMS

The simplest algorithm for solving the equations of motion is the Euler solution (Taylor series)

$$x_{n+1} = x_n + v_n \Delta t + \frac{1}{2}(F_n/m)(\Delta t)^2$$

$$v_{n+1} = v_n + (F_n/m)\Delta t$$

This does not work! It is unstable and inaccurate.

There are many sophisticated *predictor-corrector* algorithms with high order Taylor prediction and optimal correction of derivatives, but they are not better than the simple *Verlet* or *leap-frog* algorithm.

Leap Frog algorithm:

$$\begin{aligned} v_{n+\frac{1}{2}} &= v_{n-\frac{1}{2}} + (F_n/m)\Delta t \\ x_{n+1} &= x_n + v_{n+\frac{1}{2}} \Delta t \end{aligned}$$

This algorithm is

- time-reversible
- stable
- simplectic (conserves volume in (p, q) space)
- simple

PROPAGATOR TO DERIVE REVERSIBLE ALGORITHMS

[Martyna, Berne, Tuckerman]

[Review: Tuckerman and Martyna, *J. Phys. Chem.* **B 104** (2000) 159]

Use propagator in phase space to derive reversible algorithms for solving equations of motion.

Phase point $(q,p) = (q_1, \ldots, p_{3N})$ propagates as follows:

$$\frac{d}{dt}\begin{pmatrix} q \\ p \end{pmatrix} = \begin{pmatrix} \partial H/\partial p \\ -\partial H/\partial q \end{pmatrix} = i\mathcal{L}\begin{pmatrix} q \\ p \end{pmatrix}$$

$$\begin{pmatrix} q \\ p \end{pmatrix}(t) = \exp(i\mathcal{L}t)\begin{pmatrix} q \\ p \end{pmatrix}(0)$$

($\mathcal{L}$ is the *Liouville operator*, the time propagator in phase space)

For cartesian coordinates $q = x, p = v$:

$$i\mathcal{L}\begin{pmatrix} x \\ v \end{pmatrix} = \begin{pmatrix} v \\ F(x)/m \end{pmatrix} = \begin{pmatrix} v \\ o \end{pmatrix} + \begin{pmatrix} o \\ F/m \end{pmatrix} = i\mathcal{L}_v + i\mathcal{L}_F$$

$\mathcal{L}_v$ and $\mathcal{L}_F$ *do not commute*

Use Trotter split-operator technique:

$$\exp(i\mathcal{L}\Delta t) = \exp(\frac{1}{2}\, i\mathcal{L}_F\Delta t)\exp(i\mathcal{L}_v\Delta t)\exp(\frac{1}{2}\, i\mathcal{L}_F\Delta t)$$

$$= U_F(\frac{1}{2}\,\Delta t)U_v(\Delta t)U_F(\frac{1}{2}\,\Delta t)$$

PROPAGATOR TO DERIVE REVERSIBLE ALGORITHMS

Time-step propagator

$$U_F(\Delta t)\begin{pmatrix} x \\ v \end{pmatrix} = \begin{pmatrix} x \\ v + (F(x)/m)\Delta t \end{pmatrix}$$

$$U_v(\Delta t)\begin{pmatrix} x \\ v \end{pmatrix} = \begin{pmatrix} x + v\Delta t \\ v \end{pmatrix}$$

Apply $U_F(\frac{1}{2}\,\Delta t)\,U_v(\Delta t)\,U_F(\frac{1}{2}\,\Delta t)$ to $x(t_n), v(t_n)$. This yields

$$v_{n+1} = v_n + \frac{1}{2m}F_n\Delta t + \frac{1}{2m}F_{n+1}\Delta t$$

$$x_{n+1} = x_n + (v_n + \frac{1}{2m}F_n\Delta t)\Delta t$$

= Velocity Verlet algorithm ($\equiv$ Verlet $\equiv$ Leap Frog algorithms)

Verlet algorithm:

$$x_{n+1} = 2x_n - x_{n-1} + (F(x_n)/m)(\Delta t)^2$$

Leap Frog algorithm:

$$\begin{aligned} v_{n+\frac{1}{2}} &= v_{n-\frac{1}{2}} + (F_n/m)\Delta t \\ x_{n+1} &= x_n + v_{n+\frac{1}{2}}\,\Delta t \end{aligned}$$

REVERSIBLE MULTI-TIMESTEP ALGORITHMS

'RESPA': Tuckerman *et al.*, J. Chem. Phys. **93** (1990) 1287;
ibid **94** (1990) 1465; **94**(1991) 6811

Assume we can split force in contribution F_s good for short time step δt and contribution F_l good for longer time step $\Delta t = k\delta t$, then algorithm can be derived from

$$U_{F_l}(\frac{1}{2}\,\Delta t)\,U_{F_s}(\frac{1}{2}\,\delta t)\,U_v(\delta t)\;\ldots U_{F_s}(\delta t)\,U_v(\delta t)\,U_{F_s}(\frac{1}{2}\,\delta t)\,U_{F_l}(\frac{1}{2}\,\Delta t)$$

HOME WORK

---

1. Derive Velocity-Verlet algorithm from propagator
2. Prove that Verlet, Leap-Frog, and Velocity-Verlet are equivalent
3. Derive algorithm based on propagator

$$U_v(\frac{1}{2}\,\Delta t)\,U_F(\Delta t)\,U_v(\frac{1}{2}\,\Delta t)$$

GENERALIZED COORDINATES

Mass-metric tensor M

Write $\boldsymbol{r}_i$ as function of internal coordinates $q_1, \ldots q_n$.
Then kinetic energy is

$$K = \frac{1}{2} \sum_i m_i (\dot{\boldsymbol{r}}_i)^2 = \frac{1}{2} \sum_{kl} \sum_i m_i \frac{\partial \boldsymbol{r}_i}{\partial q_k} \cdot \frac{\partial \boldsymbol{r}_i}{\partial q_l} \dot{q}_k \dot{q}_l$$

in matrix notation:

$$K = \frac{1}{2} \dot{\mathbf{q}}^{\mathsf{T}} \mathbf{M} \dot{\mathbf{q}}$$

M= symmetric *mass tensor* or *mass-metric tensor*:

$$M_{kl} = \sum_i m_i \frac{\partial \boldsymbol{r}_i}{\partial q_k} \cdot \frac{\partial \boldsymbol{r}_i}{\partial q_l}$$

Compare *metric tensor* g defined by length of displacement

$$(ds)^2 = \sum_i (d\boldsymbol{r}_i)^2 = \sum_{kl} g_{kl}\, dq_k\, dq_l$$

$$g_{kl} = \sum_i \frac{\partial \boldsymbol{r}_i}{\partial q_k} \cdot \frac{\partial \boldsymbol{r}_i}{\partial q_l}$$

GENERALIZED COORDINATES

Consider a conservative system

$$\mathcal{L}(q,\dot{q}) = K(q,\dot{q}) - V(q).$$

The conjugate momenta are defined by

$$p_k = \frac{\partial K(q,\dot{q})}{\partial q_k} = \sum_l M_{kl}\dot{q}_l$$

or

$$\mathbf{p} = \mathbf{M}\dot{\mathbf{q}}$$

and the Lagrangian equations of motion are

$$\dot{p}_k = \frac{\partial \mathcal{L}}{\partial q_k} = \frac{1}{2}\dot{\mathbf{q}}^{\mathsf{T}}\frac{\partial \mathbf{M}}{\partial q_k}\dot{\mathbf{q}} - \frac{\partial V}{\partial q_k}.$$

This yields a matrix equation for $\ddot{\boldsymbol{q}}$:

$$\sum_l M_{kl}\ddot{q}_l = -\frac{\partial V}{\partial q_k} + \sum_{\alpha,\beta}\left(\frac{1}{2}\frac{\partial M_{\alpha\beta}}{\partial q_k} - \frac{\partial M_{k\alpha}}{\partial q_\beta}\right)\dot{q}_\alpha\dot{q}_\beta,$$

which has the general form

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{T}(\mathbf{q}) + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}),$$

where $\mathbf{T}$ is a generalized force or torque, and $\mathbf{C}$ is a velocity-dependent force that comprises the Coriolis and centrifugal forces. Apart from the fact that these forces are hard to evaluate, we are confronted with a set of equations that require a complexity of order n^3 to solve. Recently more efficient order-n algorithms have been devised as a result of developments in robotics.

ALTERNATIVE: CONSTRAINTS

Holonomic constraints depend only on coordinates and can be described by a *constraint equation* $\sigma(\boldsymbol{r}) = 0$ that should be satisfied at all times. For every constraint there is such an equation. Examples are (we use the notation $\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$)

- *distance constraint between two particles:* $|\boldsymbol{r}_{12}| - d_{12} = 0$, or, alternatively, $(\boldsymbol{r}_{12})^2 - d_{12}^2 = 0$.
- *angle 1-2-3 constraint between two constrained bonds:* $\boldsymbol{r}_{12}\cdot\boldsymbol{r}_{32} - c = 0$, where $c = d_{12}d_{32}\cos\phi$, or, alternatively, $\boldsymbol{r}_{13}^2 - d_{13}^2 = 0$.

Holonomic constraints are introduced into the equations of motion by minimizing the action while preserving the constraints, using Lagrange multipliers.

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}'}{\partial \dot{q}_k}\right) - \frac{\partial \mathcal{L}'}{\partial q_k} = 0 \qquad (i = 1, \cdots, n),$$

where

$$\mathcal{L}' = \mathcal{L} + \sum_{s=1}^{m} \lambda_s \sigma_s(q),$$

while for all q along the path

$$\sigma_s(q) = 0,\ s = 1, \ldots, m$$

These equations fully determine the path, i.e. both $q(t)$ and $\lambda(t)$, on a *hypersurface* determined by the constraint equations.

CONSTRAINTS-1

Practical method: **SHAKE**: Ryckaert, Berendsen, Ciccotti, *J. Comput. Phys.* **23** (1977) 327.

Shake is a program that *resets* coordinates according to the prescribed constraints.

In Verlet algorithm:

$$\boldsymbol{r}_i(t+\Delta t) = 2\boldsymbol{r}_i(t) - \boldsymbol{r}_i(t-\Delta t) + \frac{(\Delta t)^2}{m_i}\left[\boldsymbol{F}_i^u(t) + \boldsymbol{F}_i^c(t)\right],$$

where $\boldsymbol{F}^u$ are the forces disregarding the constraints, and

$$\boldsymbol{F}_i^c(t) = \sum_s \lambda_s(t)\frac{\partial \sigma_s}{\partial \boldsymbol{r}_i}$$

is the constraint force on particle i at time t. The effect of the constraint force is to add a second contribution to the displacement of the particles.

The algorithm first computes the new positions $\boldsymbol{r}_i'$ disregarding the constraints:

$$\boldsymbol{r}_i' = 2\boldsymbol{r}_i(t) - \boldsymbol{r}_i(t-\Delta t) + \frac{(\Delta t)^2}{m_i}\boldsymbol{F}_i^u(t)$$

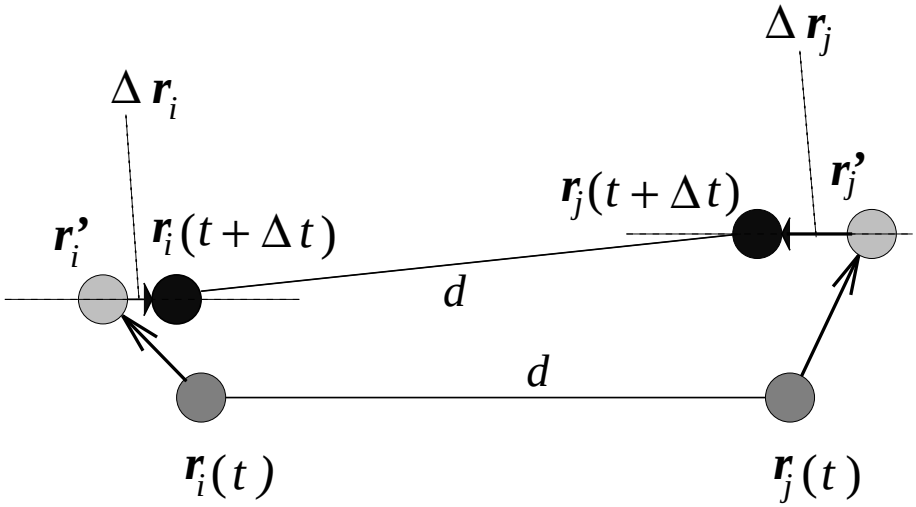
and then corrects the positions with $\Delta\boldsymbol{r}_i$ such that

$$\sigma_s(\boldsymbol{r}' + \Delta\boldsymbol{r}) = 0,\ s = 1, \ldots, m,$$

where

$$\Delta\boldsymbol{r}_i = \frac{(\Delta t)^2}{m_i}\sum_s \lambda_s(t)\frac{\partial \sigma_s(\boldsymbol{r}(t))}{\partial \boldsymbol{r}_i}$$

CONSTRAINTS-2



The set of m (generally nonlinear) coupled equations for the m λ's can be solved in several ways, always requiring iteration. They can be either linearized and then solved as a set of linear equations, or the constraints can be solved sequentially and the whole procedure iterated to convergence (= SHAKE).

A more robust and faster method **LINCS** (LINear Constraint Solver) solves a linearized matrix equation in stead
Hess, Bekker, Berendsen, Fraaije, *J. Comput. Chem.* **18** (1997) 1463

CONSTRAINTS: METRIC EFFECTS-1

Question: does constraint dynamics produce correct equilibrium averages?

Thermodynamics derived from *partition function* Q:

$$Q \propto \int dp \int dq \exp[-\beta H(p,q)]$$

$$H(p,q) = \frac{1}{2}\,\dot{\mathbf{q}}^{\mathsf{T}}\mathbf{M}\dot{\mathbf{q}} + V(\mathbf{q}) = \frac{1}{2}\,\mathbf{p}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{p} + V(\mathbf{q})$$

Integration over **p** can be carried out:

$$\int d\mathbf{p} \exp(-\frac{\beta}{2}\mathbf{p}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{p}) = (2\pi k_B T)^{3N/2} \left(\det \mathbf{M}^{-1}\right)^{-\frac{1}{2}}$$

gives

$$Q \propto \int d\mathbf{q} \,(\det \mathbf{M})^{\frac{1}{2}} \exp[-\beta V(\mathbf{q})]$$

(for cartesian coordinates $\det \mathbf{M}$ is a constant:

$$(\det \mathbf{M})^{\frac{1}{2}} = \Pi_{i=1}^{N} m_i^{3/2})$$

CONSTRAINTS: METRIC EFFECTS-2

Constraint dynamics in phase space (p', q'), with q'' constrained:

$$Q \propto \int dp' \int dq' \, \exp[-\beta H(p', q')]$$

$K = \frac{1}{2}\,\dot{\mathbf{q}}^{\mathsf{T}}\mathbf{M}\dot{\mathbf{q}}$ splits into 4 terms:

$$K = \frac{1}{2}\begin{pmatrix} \dot{\mathbf{q}}' & \dot{\mathbf{q}}'' \end{pmatrix}\begin{pmatrix} F & D \\ D^{\mathsf{T}} & C \end{pmatrix}\begin{pmatrix} \dot{\mathbf{q}}' \\ \dot{\mathbf{q}}'' \end{pmatrix}$$

$$F_{kl} = \sum_i m_i \frac{\partial \boldsymbol{r}_i}{\partial q'_k} \cdot \frac{\partial \boldsymbol{r}_i}{\partial q'_l}$$

Constraint dynamics means

$$H = \frac{1}{2}\,\dot{\mathbf{p}}'^{\mathsf{T}}\mathbf{F}^{-1}\dot{\mathbf{p}}' + V(\mathbf{q})$$

and hence

$$Q \propto \int d\mathbf{q}' \, (\det F)^{1/2} \exp[-\beta V(\mathbf{q}')]$$

in stead of

$$\int d\mathbf{q} \, (\det M)^{1/2} \exp[-\beta V(\mathbf{q})]$$

CONSTRAINTS: METRIC EFFECTS-3

In averaging over constrained simulation *extra weight factor* $w(q')$ is required:

$$w(q') = \frac{\int (\det \mathbf{M})^{1/2} dq''}{(\det \mathbf{F})^{1/2}} = \frac{(\det \mathbf{M}(q', c))^{1/2}}{(\det \mathbf{F}(q'))^{1/2}}$$

which can be expressed as an *extra potential*

$$V_c(q') = -k_B T \ln w(q')$$

This is the famous, generally neglected, often negligible, *metric tensor correction*

Fixman's theorem made computation possible
[M. Fixman, *Proc. Natl Acad. Sci.* **71** (1974) 3050]

$$(\det \mathbf{M})(\det \mathbf{Z}) = \det \mathbf{F}$$

with

$$Z_{kl} = \sum_i \frac{1}{m_i} \frac{\partial q_k''}{\partial \boldsymbol{r}_i} \cdot \frac{\partial q_l''}{\partial \boldsymbol{r}_i}$$

Hence $w(q') = (\det \mathbf{Z})^{-1/2}$

CONSTRAINTS: METRIC EFFECTS-4

Effect of metric tensor is often negligible

Example 1. *One distance constraint*:

$$q'' = r_{12} = |\boldsymbol{r}_{12}|$$

$$\det \mathbf{Z} = Z_{11} = \frac{1}{m_1}\frac{\partial r_{12}}{\partial \boldsymbol{r}_1} \cdot \frac{\partial r_{12}}{\partial \boldsymbol{r}_2} = \frac{1}{m_1} + \frac{1}{m_2}$$

$\det \mathbf{Z}$ = constant (no effect)

Example 2. *One generalised distance constraint*:

$$q'' = R = |\sum_i \alpha_i \boldsymbol{r}_i| \quad (\alpha_i \text{ constant})$$

$\det \mathbf{Z}$ = constant (no effect)

Example 3. *Two distance constraints*:

$$q_1'' = r_{12}; \quad q_2'' = r_{23}, \quad \phi = \ 1-2-3 \text{ angle}$$

$$\mathbf{Z} = \begin{pmatrix} \frac{1}{m_1} + \frac{1}{m_2} & -\frac{1}{m_2}\cos\phi \\ -\frac{1}{m_2}\cos\phi & \frac{1}{m_2} + \frac{1}{m_3} \end{pmatrix}$$

$$\det \mathbf{Z} = \left(\frac{1}{m_1} + \frac{1}{m_2}\right)\left(\frac{1}{m_2} + \frac{1}{m_3}\right) - \frac{1}{m_2^2}\cos^2\phi$$

$\det \mathbf{Z}$ depends on ϕ. Constant if ϕ is constant.

FORCE FIELDS

The potential energy description $V(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N)$ contains

- *bonded interactions* (fixed list), as
  - bond lengths (e.g. harmonic, or constraint)
  - bond angles (harmonic)
  - dihedral angles (periodic)
  - improper dihedrals (keeping groups flat or preventing mirror image)
- *non-bonded interactions* (dynamics list), as
  - Coulomb interactions (full or partial charges, dipoles, multipoles)
  - dispersion interactions (attractive, $\propto r^{-6}$)
  - repulsive interactions (short-range, $\propto r^{-12}$)

With *periodic boundary conditions* Coulomb interaction should be computed with *lattice sums* (Ewald, 1921). Best method is Particle mesh Ewald by Darden, York and Pederson, *J. Chem. Phys.* **98** (1993) 10089.

Use of *cut-off radius* introduces errors and noise. Acceptable is use of cut-off for neutral *charge groups*, enhanced with *reaction field* to account for dielectric environment.

POLARIZABILITY

Present-day force fields still lack *polarizability*: dipoles induced by electric field. This is a *non-pairadditive* interaction. Effects are

- no electrostatic shielding due to high-frequency dielectric constant
- small molecule models (water) have enhanced dipoles (representing average polarization in the liquid)
- models use *effective pair potentials*, good for one environment only
- second virial coefficient of the gas phase is twice the experimental value (water)
- solvation energy of ions in nonpolar environment is neglected (e.g. Cl^- in a protein)
- ligand binding energy of divalent ions is underestimated (e.g. Ca^{2+} in protein liganded with water or carbonyl oxygen)
- polar/nonpolar partition likely to be wrong.

SIMULATION BOX

- **Isolated cluster**
  advantages: simple; solvent can be limited to shell or omitted and replaced by effective boundary potential,
  disadvantages: strong boundary effects (outer layer must be discarded); very wrong electrostatic interactions unless *reaction field* is imposed.

- **Periodic box**
  advantages: avoids boundary effects; mimics infinite environment; consistent treatment of long-range interactions possible,
  disadvantages: periodicity is artefact, effect must be evaluated (especially Coulombic artefacts).

  Triclinic box (unit cell: $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$) is universal and can hold all shapes, e.g. near-spherical *truncated octahedron*. Box shape and size can be optimized to minimize number of solvent molecules for any macromolecular shape.

TEMPERATURE AND PRESSURE CONTROL-1

- **No controls**
  Without controls, MD should generate a *microcanonical ensemble* (N, V, E constant). However, integration errors, force fluctuations, and inconsistencies in the forces (e.g. by using a cut-off radius) cause fluctuations and slow drifts in total energy.

  Systems that are not in equilibrium will go to equilibrium while the temperature changes. We often prefer a N, V, T, or N, p, T ensemble or wish to control temperature in a prescribed, time-dependent scheme.

- **Extended system control**
  Extend system with extra degree(s) of freedom that allow to keep temperature and/or pressure near a prescribed value (*Nosé-Hoover thermostat*).
  Advantage: the system retains a Boltzmann equilibrium distribution,
  Disadvantage: the extra degrees of freedom have an artificial 'mass'; the response of the system to temperature or pressure changes is oscillatory; the ergodicity has been questioned.

TEMPERATURE AND PRESSURE CONTROL-2

- **Weak coupling to a bath**
  Modify equations of motion such that system temperature or pressure approaches the required ('bath') temperature or pressure with a given time constant.
  Advantages: the response of the system is first-order exponential; coupling is flexible, from strong (fast response) to weak (negligible influence on system behaviour),
  Disadvantages: no known ensemble is generated with intermediate coupling constants; overall fluctuations cannot be used.

- **Constraining to specified T and p**
  Change velocities and volume at each step to set temperature and pressure exactly at prescribed values.
  Advantages: immediate response; known ensemble,
  Disadvantages: dynamics influenced in unknown manner.