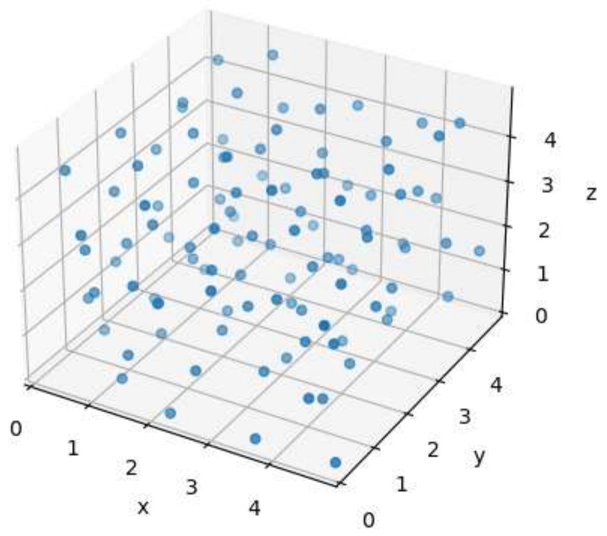
分子动力学



$$\mathbf{F}_i = m_i \mathbf{a}_i$$

Force $\boldsymbol{F}$ can be obtained by first-principles methods or force field methods

$$m \frac{\mathrm{d}^2 \mathbf{r}_i}{\mathrm{d}t^2} = -\nabla_i V(\{r_i\})$$

积分算法

- Euler algorithm: simplest but worst

$$\mathbf{r}_i(t+\Delta t) = \mathbf{r}_i(t) + \mathbf{v}_i(t)\Delta t + O(\Delta t^2)$$

- Verlet algorithm (Verlet, 1967)

$$\begin{cases} \mathbf{r}_i(t+\Delta t) = \mathbf{r}_i(t) + \mathbf{v}_i(t)\Delta t + \dfrac{\mathbf{F}_i(t)}{2m}\Delta t^2 + \dfrac{\dddot{\mathbf{r}}_i}{6}\Delta t^3 + O(\Delta t^4) \\ \mathbf{r}_i(t-\Delta t) = \mathbf{r}_i(t) - \mathbf{v}_i(t)\Delta t + \dfrac{\mathbf{F}_i(t)}{2m}\Delta t^2 - \dfrac{\dddot{\mathbf{r}}_i}{6}\Delta t^3 + O(\Delta t^4) \end{cases}$$

$$\Rightarrow \quad \mathbf{r}_i(t+\Delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t-\Delta t) + \frac{\mathbf{F}_i(t)}{m}\Delta t^2 + O(\Delta t^4) \qquad \Delta t \sim 1\text{ - }10\text{ fs}$$

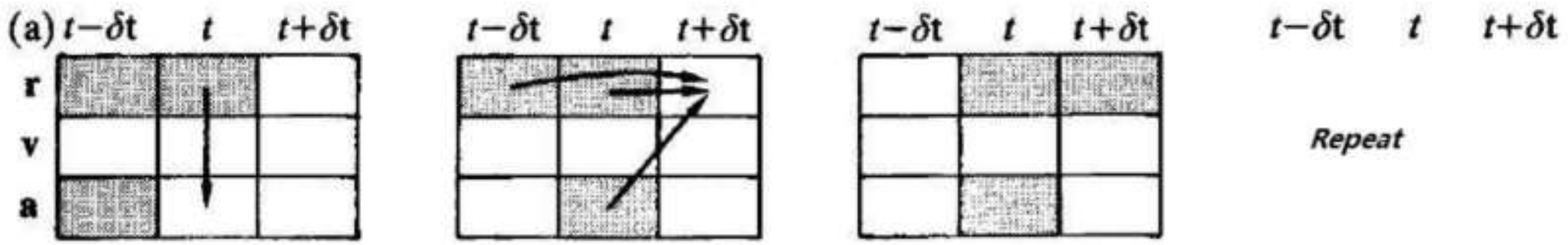
$$\mathbf{v}(t) = \frac{\mathbf{r}(t+\delta t) - \mathbf{r}(t-\delta t)}{2\delta t} + O(\Delta t^2)$$

积分算法

- Verlet algorithm (Verlet, 1967)

$$\mathbf{r}_i(t+\Delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t-\Delta t) + \frac{\mathbf{F}_i(t)}{m}\Delta t^2 + O(\Delta t^4) \qquad \Delta t \sim 1\text{ - }10\text{ fs}$$

$$\mathbf{v}(t) = \frac{\mathbf{r}(t+\delta t) - \mathbf{r}(t-\delta t)}{2\delta t} + O(\Delta t^2)$$



其中t表示时间，r表示位置，v表示速度，a表示加速度。

积分算法

- velocity Verlet (Swope, 1982)

$$\mathbf{r}(t+\delta t)=\mathbf{r}(t)+\mathbf{v}(t)\delta t+\frac{1}{2}\mathbf{a}(t)\delta t^2+\mathrm{O}\ (\delta t^3)$$

$$\mathbf{v}(t+\delta t)=\mathbf{v}(t)+\boldsymbol{a}(t)\delta t+\frac{1}{2}\dot{\mathbf{a}}(t)\delta t^2+\mathrm{O}\ (\delta t^3)$$

$$\Downarrow$$

$$\mathbf{v}(t+\delta t)=\mathbf{v}(t)+\boldsymbol{a}(t)\delta t+\frac{1}{2}\frac{\boldsymbol{a}(t+\delta t)-\boldsymbol{a}(t)}{\delta t}\delta t^2+\mathrm{O}\ (\delta t^3)$$

$$\mathbf{v}(t+\delta t)=\mathbf{v}(t)+\frac{1}{2}[\mathbf{a}(t)+\mathbf{a}(t+\delta t)]\delta t$$

积分算法

- velocity Verlet (Swope, 1982)

$$\mathbf{r}(t+\delta t)=\mathbf{r}(t)+\mathbf{v}(t)\delta t+\frac{1}{2}\mathbf{a}(t)\delta t^2+O\left(\delta t^3\right)$$

$$\mathbf{v}(t+\delta t)=\mathbf{v}(t)+\frac{1}{2}[\mathbf{a}(t)+\mathbf{a}(t+\delta t)]\delta t$$

速度计算相当于分成两步,半步速度

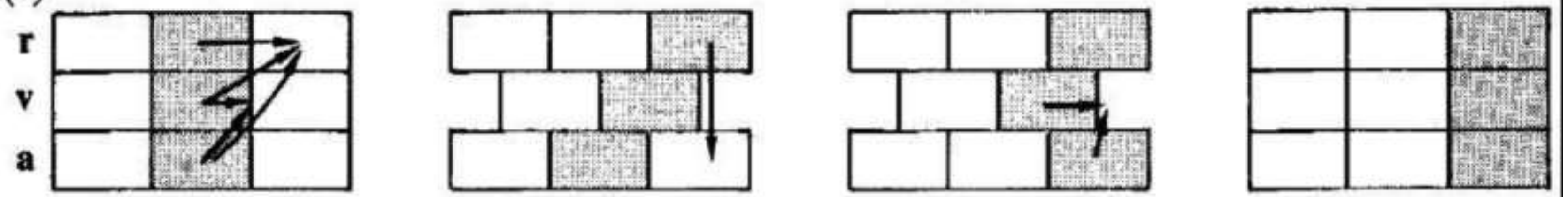
$$\mathbf{v}\left(t+\frac{1}{2}\delta t\right)=\mathbf{v}(t)+\frac{1}{2}\mathbf{a}(t)\delta t$$

$$\mathbf{r}(t+\delta t)=\mathbf{r}(t)+\mathbf{v}\left(t+\frac{1}{2}\delta t\right)\delta t$$

$$\mathbf{v}(t+\delta t)=\mathbf{v}\left(t+\frac{1}{2}\delta t\right)+\frac{1}{2}\mathbf{a}(t+\delta t)\delta t$$

1. Calculate $\mathbf{v}\left(t+\frac{1}{2}\,\Delta t\right)=\mathbf{v}(t)+\frac{1}{2}\,\mathbf{a}(t)\,\Delta t$.
2. Calculate $\mathbf{x}(t+\Delta t)=\mathbf{x}(t)+\mathbf{v}\left(t+\frac{1}{2}\,\Delta t\right)\,\Delta t$.
3. Derive $\mathbf{a}(t+\Delta t)$ from the interaction potential using $\mathbf{x}(t+\Delta t)$.
4. Calculate $\mathbf{v}(t+\Delta t)=\mathbf{v}\left(t+\frac{1}{2}\,\Delta t\right)+\frac{1}{2}\,\mathbf{a}(t+\Delta t)\Delta t$.

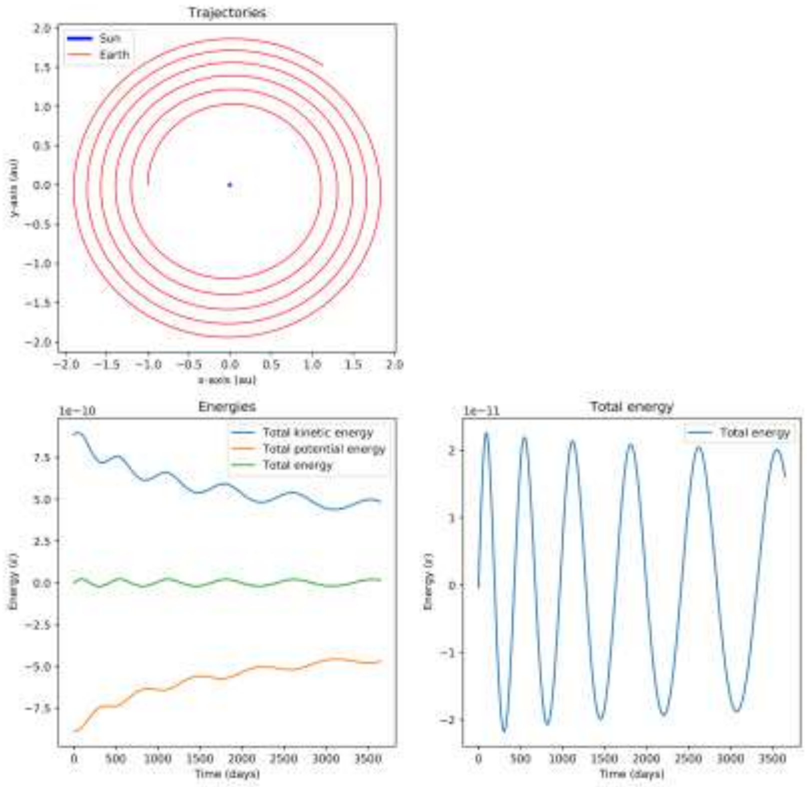
(c)



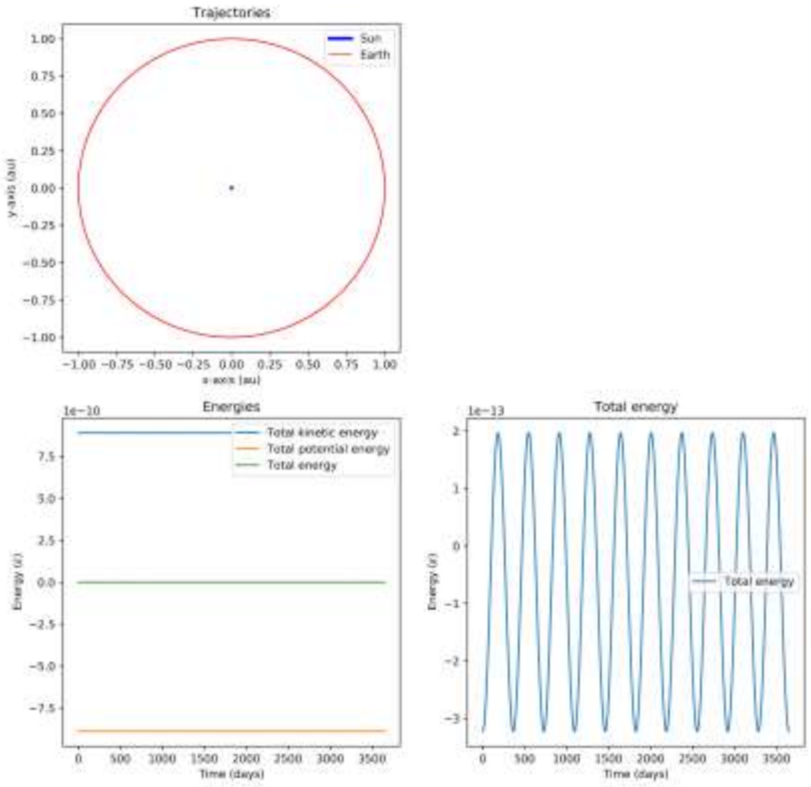
1. Calculate $\mathbf{x}(t+\Delta t)=\mathbf{x}(t)+\mathbf{v}(t)\,\Delta t+\frac{1}{2}\,\mathbf{a}(t)\,\Delta t^2$.
2. Derive $\mathbf{a}(t+\Delta t)$ from the interaction potential using $\mathbf{x}(t+\Delta t)$.
3. Calculate $\mathbf{v}(t+\Delta t)=\mathbf{v}(t)+\frac{1}{2}\,\big(\mathbf{a}(t)+\mathbf{a}(t+\Delta t)\big)\Delta t$.

积分算法

Euler's method



Velocity-Verlet's method



积分算法

1. “Good” algorithm means: energy conservation, time reversibility, volume preservation in phase space
2. Time step: as large as possible until energy is not conserved, usually ~ 1 - 10 fs
3. Lyapunov instability? Ergodicity?

Lyapunov instability: 随模拟时间的增加，模拟轨迹以指数方式偏离实际的轨迹

积分算法

◆ 时间反演(time reversible)

- Verlet, Velocity Verlet are time reversible
- Beeman, Velocity is not time reversible

◆ 能量守恒(energy conservation)

- 短时间

  低阶方法（如Verlet）可能短时间尺度内能量涨落比较大
- 长时间

  高阶方法能量可能会漂移

Ensembles

1. Microcanonical ensemble: NVE

   Particle number (N)–volume(V)– energy (E) are conserved

2. Canonical ensemble: NVT

   Particle number (N)–volume(V) are conserved @ given (average) temperature

3. NPT: fixed particle number, Pressure and temperature

Thermostat

Instantaneous Temperature:

$$\frac{3}{2} N_{at} k_B T_c = \frac{1}{2} \sum_i m_i \mathbf{v}_i^2 \quad \Rightarrow \quad T_c = \frac{\sum_i m_i \mathbf{v}_i^2}{3 N_{at} k_B}$$

1. Velocity Rescaling $\quad \mathbf{v}'_i = \sqrt{T/T_c}\, \mathbf{v}_i$

Velocity scaling schemes do not strictly follow the canonical ensemble, though in practice, the amount they deviate from canonical is quite small. It leads to discontinuities in the momentum part of the phase space trajectory due to the rescaling procedure at each time step.

Thermostat

2. Nosé-Hoover Thermostat

The Nosé-Hoover thermostat is a deterministic algorithm for constant temperature molecular dynamics simulations.

Originally developed by Nosé and was improved further by Hoover. Although the heat bath of Nosé-Hoover thermostat consists of only one imaginary particle, simulation systems achieve realistic constant-temperature condition (canonical ensemble). The Nosé–Hoover thermostat has been commonly used as one of the most accurate and efficient methods for constant-temperature molecular dynamics simulations.

Thermostat

Introduce a friction variable $\xi(t)$ as “heat bath”

$$\begin{cases} \dot{\mathbf{q}}_i = \dfrac{\mathbf{p}_i}{m_i} \\ \dot{\mathbf{p}}_i = \mathbf{F}_i - \xi(t)\mathbf{p}_i \\ Q\dot{\xi}(t) = \displaystyle\sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i} - \frac{3}{2} N k_\mathrm{B} T_0 \end{cases}$$

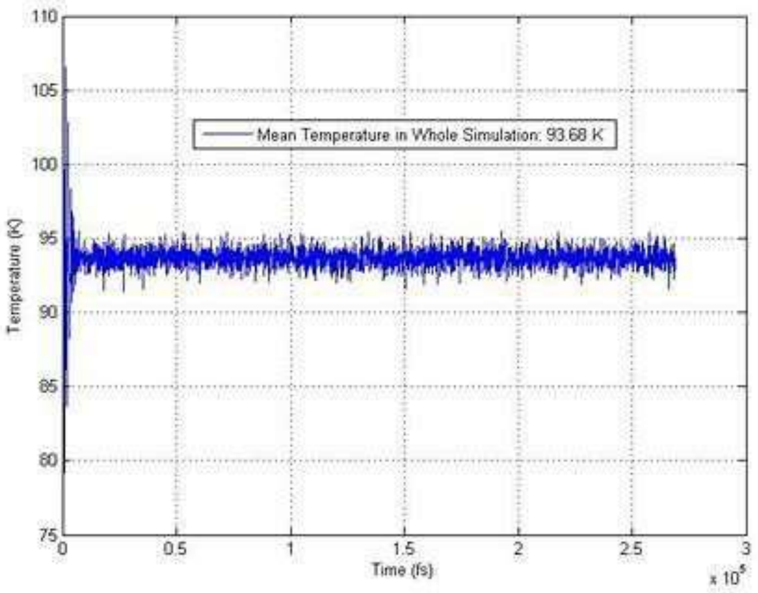
Q is the “heat bath” mass. Large Q is weak coupling.
The value of this parameter must be set carefully, because if it is chosen to be too small, the phase space of the system will not be canonical, and it is chosen to be too large the temperature control will not be efficient.

Thermostat

The Hamiltonian of the system + heat bath is:

$$H_{NH} = H(\mathbf{p}, \mathbf{q}) + Q\frac{\xi^2}{2} + N_{df}k_B T_0 q_\xi \qquad q_\xi = \int \xi(t)dt$$

The energy of the physical system fluctuates, but the energy of the physical system plus the heat bath is conserved.



Barostat

Instantaneous Pressure:

$$\frac{\beta P}{\rho} = 1 - \frac{\beta}{3N}\left\langle \frac{1}{2}\sum_{i,j} r_{ij}\frac{\partial U}{\partial r_{ij}} \right\rangle$$

Berendsen barostat

$$V_i^{new} = V_i^{old} \cdot \lambda$$

$$r_i^{new} = r_i^{old} \lambda^{1/3}$$

$$\frac{\partial P(t)}{\partial t} = \frac{1}{\tau}(P_{bath} - P(t))$$

$$\lambda = \left(1 - \frac{k\delta t}{\tau}(P(t) - P_{bath})\right)$$

原子间的相互作用势

两体势： Lennard-Jones (LJ)，Morse

惰性气体、简单金属

原子内嵌势： Embed Atom Methods (EAM)

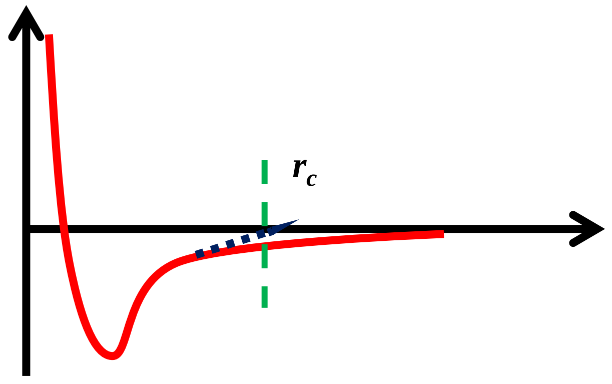
简单金属、过渡金属

紧束缚势： Tersoff, Brennerd, Stillinger-WeberPotential

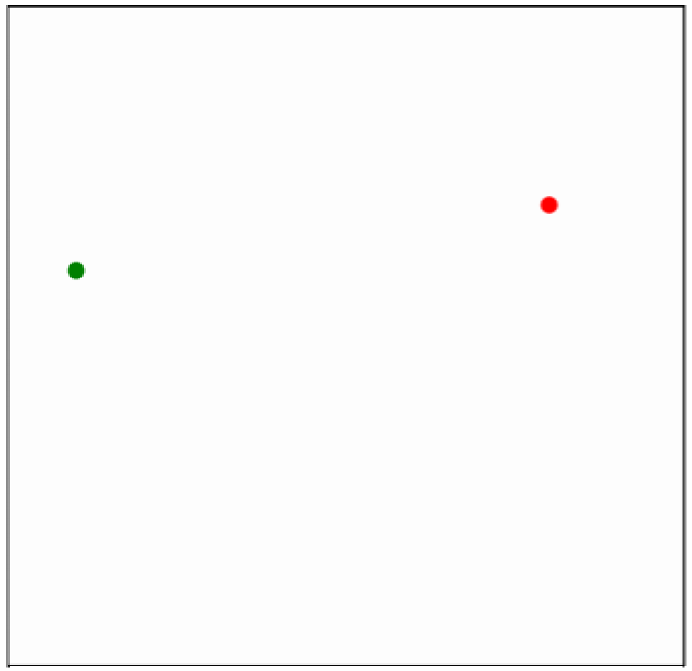
半导体

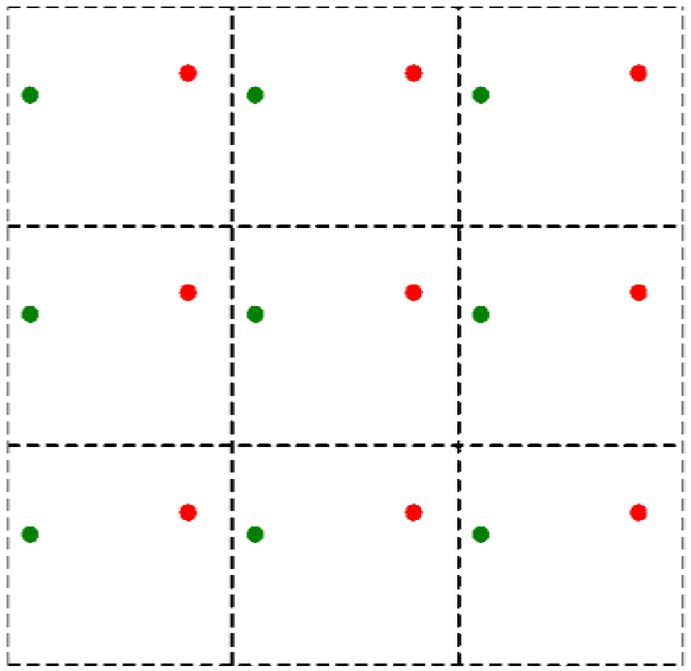
Lennard-Jones (LJ)势

$$V = 4\epsilon\left(\frac{\sigma^{12}}{r^{12}} - \frac{\sigma^{6}}{r^{6}}\right)$$



周期性边界条件



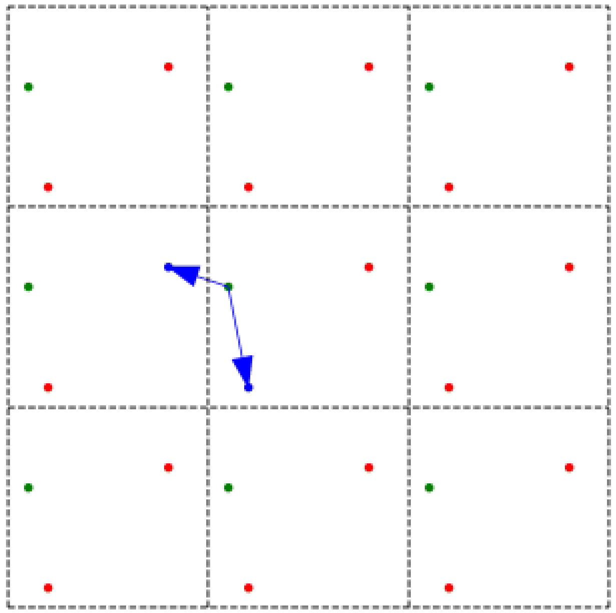


- Macroscopic systems $O(10^{23})$ particles.
- We eliminate *surfaces* from simulation.
- Allows us to get at *bulk properties with few particles*.
- Applied to solids, liquids, gases, and plasmas.
- Some finite-size effects remain, but can often be removed with *scaling*.

周期性边界条件

Minimum Image Convention: **Take the closest distance $|r|_M = \min(r+nL)$**

$$\begin{aligned} -L/2 < x_i - x_j < L/2 &\rightarrow x'_j = x_j \\ x_i - x_j > L/2 &\rightarrow x'_j = x_j + L \\ x_i - x_j < -L/2 &\rightarrow x'_j = x_j - L \end{aligned}$$



How to handle the "tail" of the potential:

- *Potential cutoff* : **$V(r)=0$ for $r > L/2$**
  - continuous potential: **$V_c(r) = V(r) - V(r_c)$** for $r < r_c$.
  - continuous V&F: **$V_c(r) = V(r) - V(r_c) - \nabla V(r_c)*(r-r_c)$** for $r < r_c$.
- *Image potential*: **$V_I = \Sigma\, v(r_i-r_j + nL)$**

  For long-range potential this leads to the *Ewald image potential*.

  You need a charge background and convergence factor (more later)

LJ Force Computation

```
! Loop over all pairs of atoms.
      do i=2,natoms
        do j=1,i-1
!Compute distance between i and j.
          r2 = 0
    do k=1,ndim
            dx(k) = r(k,i) - r(k,j)
!Periodic boundary conditions.
          if(dx(k) .gt. cell(k)/2) dx(k) = dx(k)-cell(k)
          if(dx(k) .lt. cell(k)/2) dx(k) = dx(k)+cell(k)
            r2 = r2 + dx(k)*dx(k)
     enddo
```

We can do better with the PBC!
    dx(k)=dx(k)-cell(k)*nint(dx(k)/cell(k))

Why?

```
!Only compute for pairs inside radial cutoff.
          if(r2.lt.rcut2) then
            r2i=sigma2/r2
            r6i=r2i*r2i*r2i
!Shifted Lennard-Jones potential.
            pot = pot+eps4*r6i*(r6i-1)- potcut
!Radial force.
            rforce = eps24*r6i*r2i*(2*r6i-1)
            do k = 1 , ndim
              force(k,i)=force(k,i) + rforce*dx(k)
              force(k,j)=force(k,j) - rforce*dx(k)
            enddo
          endif
        enddo
      enddo
```

- Complexity is defined as how a computer algorithm scales with the number of degrees of freedom (atoms).

- Number of terms in pair potential is $N(N-1)/2 \approx O(N^2)$

- ***For short-range*** V(r), use *neighbor tables* to reduce it to O(N).
  - ***Explicit Neighbor list (Verlet)*** for systems that move slowly (keep a list and update occassionally.)
  - ***Bin Sort list*** (sort particles into bins and sum over neighboring bins)

- ***Long-range potentials*** require more sophisticated treatment:
  - *Ewald sums* are $O(N^{3/2})$
  - *Fast Multipole Algorithms* are O(N), for very large N.
  - Tree Methods

- More later…

结构优化

- Find a local minimum on PES

$$f(\vec{x}) = a + \vec{b}\vec{x} + \frac{1}{2}\vec{x}\mathbf{B}\vec{x} = \bar{a} + \frac{1}{2}(\vec{x} - \vec{x}^0)\mathbf{B}(\vec{x} - \vec{x}^0)\ \mathbf{B}_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

- Gradient

$$g(x) = \frac{\partial f}{\partial x} = \mathbf{B}\left(x - x^0\right)$$

- Newton’s method (find the root of the derivative)

$$x = x^1 - \mathbf{B}^{-1} g(x^1)$$

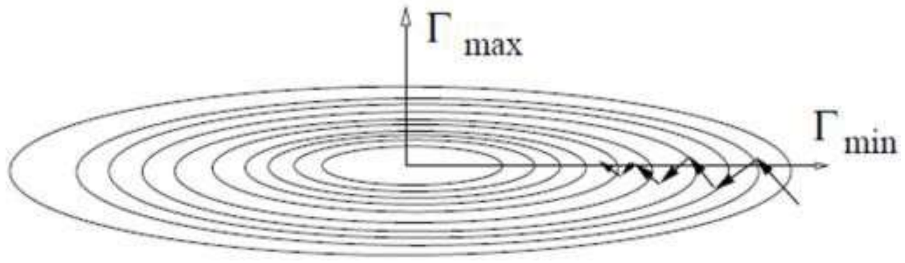
结构优化

Steepest Descent (SD)

- Guess x^1
- Calculate $g(x^1)$
- Step along the steepest descent direction

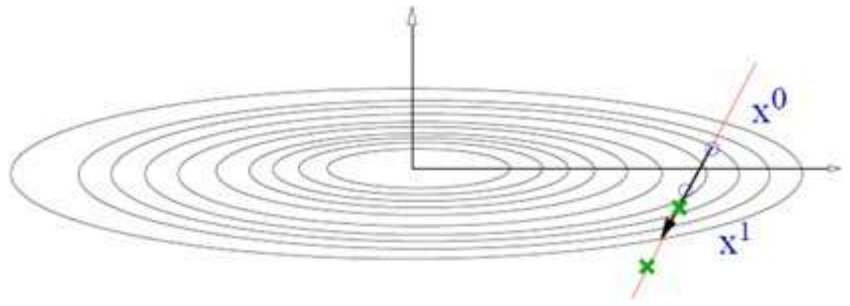
$$x^2 = x^1 - \frac{1}{\Gamma_{\max}} g(x^1)$$

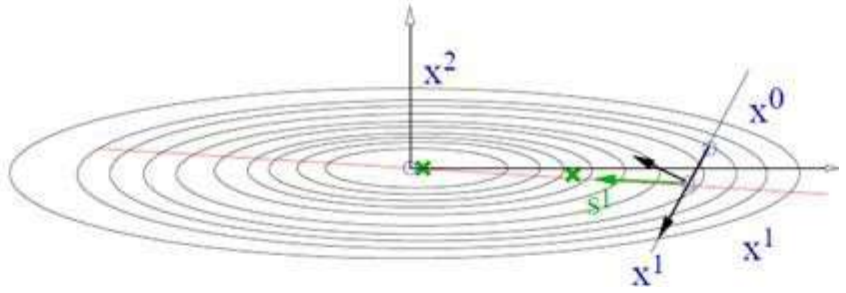
- Repeat to get converged geometry



Conjugate Gradient (CG)

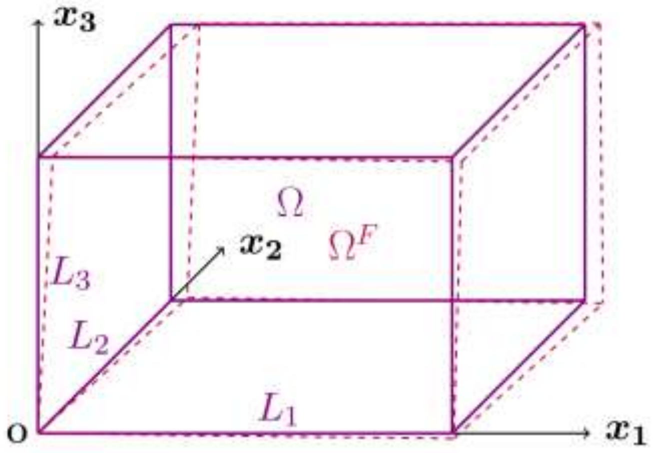
- The CG Algorithm
  - SD from x^0 along g^0
  - Line minimization to x^1
  - New gradient $g^1 = g(x^1)$ and conjugated gradient s^1
  - Points directly towards the minimum
- Line minimization is done using a variant of Brent algorithm
  - trial step along search direction (conjg. gradient scaled by POTIM)
  - quadratic or cubic interpolation using energies and forces at x_0 and x_1 allows to determine the approximate minimum
  - continue minimization as long as approximate minimum is not accurate enough





晶格优化

- Optimization by hand: do a series of calculations at different volumes
- Optimization by hand: fit to thermodynamics equations
- Automatic optimizations based on Hellmann-Feynman stresses
  - Selection of Initial Structure and Calculation of Hellmann-Feynman Stresses
  - Update Lattice Parameters: Based on the calculated stresses, adjust the lattice constants and atomic positions to reduce the total energy of the system.
  - Iterative Optimization: Repeat the stress calculations and structure adjustments until the stresses meet the convergence criteria and the system's total energy reaches a minimum.



Hellmann-Feynman stress tensor can be defined as:

$$\sigma_{\alpha\beta} = \frac{1}{|\Omega|} \frac{\partial \mathcal{L}^F(\mathbf{\Psi}, \mathbf{g}, \phi, \mathbf{R}^F)}{\partial F_{\alpha\beta}}\bigg|_{\mathcal{G}}, \quad \alpha, \beta \in \{1, 2, 3\}$$