

Molekuladinamika

Számítógépes szimulációk

szamszimf17la

Csabai István, Stéger József

ELTE

Komplex Rendszerek Fizikája Tanszék

Email: csabai@complex.elte.hu, steger@complex.elte.hu

A **molekuladinamika (MD)** nagy számú részecske mikroszkopikus dinamikájának kiértékelésével foglalkozik.

Mai számítógépekkel a makroszkopikus rendszerekre jellemző részecskeszám ($6 \cdot 10^{23}$ nagyságrend) mikroszkopikus mozgását nem lehetséges követni.

Cél, hogy minél több részecskét, minél pontosabban szimuláljunk, amelyekkel már a rendszer termodinamikai tulajdonságai vizsgálhatóak.

Ha az erő elég gyorsan lecseng, akkor a $\sim N^2$ párkölcsönhatás számolásánál közelítéseket vezetünk be:

- a távoli erők nagyobb térrészekre kiátlagolhatóak,
- a *nagyon* távol eső részecskék pedig elhanyagolhatóak.

Megjegyzés: A hosszú hatótávolságú erőknél (pl. a gravitáció) az efajta elhanyagolás nem lehetséges, más közelítéseket kell alkalmazni, amellyel az ***N*-test szimuláció** problémaköre foglalkozik.

- Adott N atom, melyek mindegyikének ismert a kezdő sebessége és pozíciója. (Ez tipikusan nemegyensúlyi helyzet.)
- A Newton-törvények értelmében a részecskék közt páronként számoljuk ki az erőt, és időben léptessük a részecskéket.
- Figyeljük, hogy a rendszer a kezdeti tranziens után termodinamikai egyensúlyba kerül-e.

Egyensúlyban teljesül az ekvipartíció tétele

$$\langle K \rangle = \left\langle \frac{1}{2} m v^2 \right\rangle = \frac{3}{2} k_B T,$$

ahol $k_B = 1,38 \cdot 10^{-23} \frac{\text{J}}{\text{K}}$ a Boltzmann-állandó.

A szimuláció során $\langle K \rangle$ mutatja, hogy egyensúlyban van-e a rendszer. Egyensúlyban mérhetőek a termodinamikai változók: a hőmérséklet (T), a nyomás (p), a hőkapacitás (C_V), stb.

Az atomok belső szerkezetét is figyelembe vevő *kvantummechanikát* kellene használni, de a nemesgázok szimulációja során a *klasszikus részecske közelítés* elfogadható eredményre vezet.

- ❶ A $T = 300$ K hőmérsékletű Ar gázban az egy részecskére jutó mozgási energia

$$\frac{3}{2} k_B T \cong 6,2 \cdot 10^{-21} \text{ J} = 0,039 \text{ eV}$$

sokkal kisebb, mint az elektronok gerjesztési energiája (11,6 eV – zárt elektronhéj).

- ❷ Az $m = 6,69 \cdot 10^{-26}$ kg tömegű Ar atomok de Broglie-hullámhossza

$$\frac{2\pi\hbar}{p} = \frac{2\pi\hbar}{\sqrt{3mk_B T}} \cong 2,2 \cdot 10^{-11} \text{ m}$$

kiseb mint az atom effektív mérete, és így az atomok közötti tipikus távolságnál is ($r_0 = 3,4 \cdot 10^{-10}$ m).

Probléma lehet a termodinamikai egyensúly elérése.

A szimulációs lépéshosszt kisebbnek kell választani, mint az atomok közötti átlagos távolság megtételéhez szükséges idő.

Az Ar atomok tipikus átlagos sebességével számolva:

$$\tau \approx \frac{r_0}{\sqrt{3k_B T/m}} \cong 7,9 \cdot 10^{-13} \text{ s},$$

ami kevesebb mint 1 picosecundum. $N = 10^3$ atomra minden lépésben minden erőt kiszámolva, néhány nanosecundum szimulációja akár órákig is tarthat.

Ezért fontos a kezdeti feltételeknek a termodinamikai egyensúlyhoz minél közelebbi beállítása.

A legelterjedtebb MD differenciálegyenlet léptető algoritmusok:

- a Verlet-algoritmus,
- a velocity-Verlet-algoritmus.

Jelölésrendszer

Az n -ik szimulációs lépésben a részecskék

- koordinátái $\vec{R}_n = (\mathbf{r}_1, \dots, \mathbf{r}_N)_n$,
- a sebességeik \vec{V}_n és
- a gyorsulások \vec{A}_n .

A Verlet-algoritmus léptető szabályai

$$\begin{aligned}\vec{R}_{n+1} &= 2\vec{R}_n - \vec{R}_{n-1} + \tau^2 \vec{A}_n + \mathcal{O}(\tau^4), \\ \vec{V}_n &= \frac{\vec{R}_{n+1} - \vec{R}_{n-1}}{2\tau} + \mathcal{O}(\tau^2).\end{aligned}$$

A módszer előnyei:

- Gyorsabb mint a Runge-Kutta-módszer: csak egyszer kell kiszámolni a gyorsulásokat egy lépés során.
- Majdnem olyan pontos, mint a negyedrendű Runge-Kutta-módszer, ahol a hiba $\mathcal{O}(\tau^5)$ rendű.
- Jól megőrzi az energiát, ami nagyon fontos a sokrészecske dinamikánál.
- Ergodikus: időtükrözésre nem változik, ami fontos a részletes egyensúly feltétele miatt.

A módszer hátrányai:

- *Három pontos rekurzió*: azaz két előző lépést használ, így nem indítható egy kezdeti feltételből.
- A sebesség hibája $\mathcal{O}(\tau^2)$, bár sebességtől független erők esetén, ez nem rontja a pozíciók pontosságát.
- A sebesség és a pozíció nem egyszerre frissül, a sebesség *lemarad*.

A velocity-Verlet-algoritmus léptetési szabályai

$$\begin{aligned}\vec{R}_{n+1} &= \vec{R}_n + \tau \vec{V}_n + \frac{\tau^2}{2} \vec{A}_n + \mathcal{O}(\tau^3), \\ \vec{V}_{n+1} &= \vec{V}_n + \frac{\tau}{2} (\vec{A}_{n+1} + \vec{A}_n) + \mathcal{O}(\tau^3).\end{aligned}$$

A módszer előnyei:

- Nem követel meg időben két korábbi lépést.
- Egyszerre frissíti a koordinátákat és sebességeket.
- A sebesség pontosabb.

A módszer hátrányai:

- \vec{R} -ben csak $\mathcal{O}(\tau^3)$ rendben pontos.

A részecskék közötti kölcsönhatást a van der Waals-közelítésben a Lennard-Jones-potenciál írja le.

A potenciál

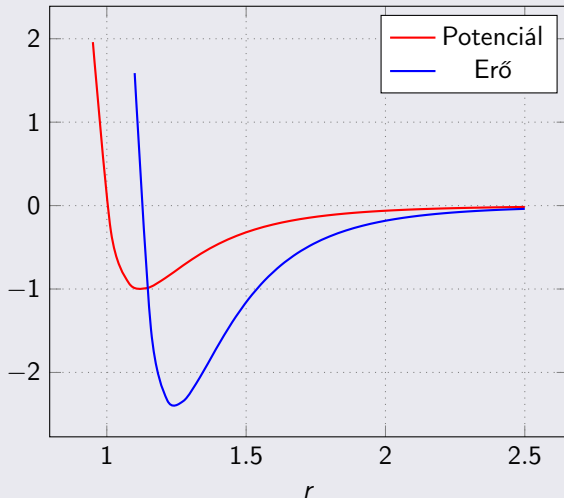
Az atomok középpontjai közötti távolság r

$$V(r) = 4V_0 \left[\left(\frac{r_0}{r} \right)^{12} - \left(\frac{r_0}{r} \right)^6 \right].$$

A Lennard-Jones-potenciál tulajdonságai

- $r = r_0$ -nál előjelet vált,
- $r = 2^{1/6} r_0$ -ben veszi fel minimumát, ami $-V_0$,
- erő erősen taszító kis távolságokon: zárt elektronhéjak Pauli-féle kizárási elv,
- gyengén vonzó nagy távolságokon: van der Waals-kölcsönhatás miatt indukált elektromos dipólerők.

A Lennard-Jones-potenciál $r_0 = 1$ és $V_0 = 1$ esetén:



A potenciál gradienseiből számolható az erő:

$$\vec{F}(\mathbf{r}) = -\underline{\nabla} V(r) = \frac{24V_0}{r^2} \left[2 \left(\frac{r_0}{r} \right)^{12} - \left(\frac{r_0}{r} \right)^6 \right] \mathbf{r}.$$

A rendszer energiája:

$$E = \sum_{i=1}^N \frac{1}{2} m v_i^2 + \sum_{\langle ij \rangle} V(r_{ij}),$$

ahol r_{ij} az $\langle ij \rangle$ atompár közötti távolság.

Egyensúlyban a sebességek a Maxwell-Boltzmann-eloszlást követik.

Modellezzük N darab Ar atom viselkedését egy négyzetben.

- A LJ-potenciál paraméterei argonra $V_0 = 1,65 \cdot 10^{-21}$ J, azaz $V_0/k_B = 119,8$ K és $r_0 = 3,41 \cdot 10^{-10}$ m.
- A rendszer karakterisztikus ideje

$$\tau_0 = \sqrt{\frac{mr_0^2}{V_0}} = 2,17 \cdot 10^{-12} \text{ s.}$$

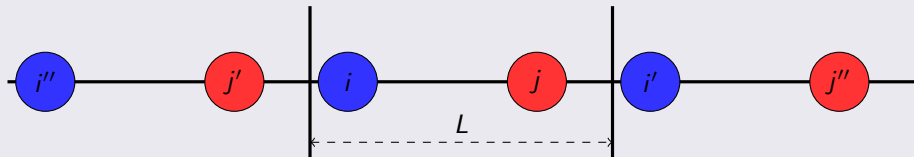
Numerikus szempontból nem praktikus SI értékekkel szimulálni.

Válasszuk meg a skálát úgy, hogy $m = 1$, $r_0 = 1$, $V_0 = 1$ és $\tau_0 = 1$.

Modellezzük N darab argon atom viselkedését egy négyzetben (folytatás).

- A határokon ne kelljen speciális feltételekkel foglalkozni (fallal ütközés): tórusz topológia.
- Gondoskodni kell arról is, hogy ha egy részecske az egyik oldalon elhagyja a cellát, akkor szemközti oldalon visszakerüljön.
- A legnagyobb erőhatást követeljük meg: egy adott részecskéhez a pár lehetséges képei közül mindig a legközelebbi példányt vesszük. Ez nem mindig az aktuális cellában van, pl $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j| > r_{ij'} = |\mathbf{r}_i - \mathbf{r}_{j'}|$.
- Ha nem így járnánk el, akkor a pár képeinek a hatásai fizikailag értelmetlen korrelációkat hozna be.

A periodikus határfeltétel érzékeltetése 1D-ban



Kezdőfeltételek beállítása (tranziens időtartamának minimalizálása):

- ❶ Kondenzált állapotban az atomok általában háromszögrácsban helyezkednek el, ez minimalizálja a potenciális energiát. *Ehelyett egyszerű négyzetrácsot használunk.*
- ❷ A sebességek egyensúlyi eloszlása:

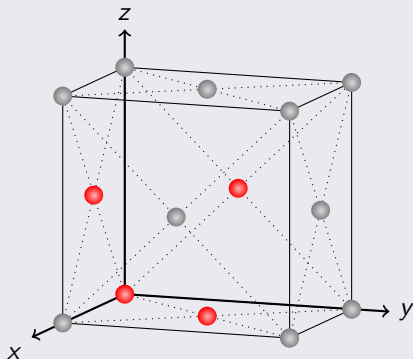
$$\mathcal{P}(v)dv = \left(\frac{m}{k_B T} \right) e^{-\frac{mv^2}{2k_B T}} v dv.$$

Ehelyett egy adott intervallumban eloszló egyenletes véletlen értékekről indítjuk a sebességeket.

A három dimenziós probléma modellezése:

- periodikus határfeltétel,
- a potenciális energia minimumra jellemzően a rendszer a lapcentrált köbös elrendezésből indítva

A lapcentrált köbös elemi cella



- Jó kitöltés feltétele: $N = 4M^3$
(pl. 32, 108, 256, 500...)

- Az atomok pozíciói:

$$(0; 0; 0), (0, 5; 0, 5; 0), \\ (0, 5; 0; 0, 5), (0; 0, 5; 0, 5),$$

Eltolva $(0, 25; 0, 25, 0, 25)$ -tel, hogy ne a határon legyenek.

Az elemi cella 4 atomot tartalmaz.

A három dimenziós probléma modellezése (folytatás):

- 3D-ben a Maxwell-Boltzmann-eloszlás a Gauss-eloszlás:

$$\mathcal{P}(\mathbf{v}) = \left(\frac{m}{2\pi k_B T} \right)^{3/2} e^{-\frac{m(v_x^2 + v_y^2 + v_z^2)}{2k_B T}}.$$

- 1 A kódban a *Numerical Recipes*-ből átvett `gasdev()` függvény 0 várható értékű, 1 szórású véletlen számokat állít elő egyenletes eloszlásokból a [Box-Müller-algoritmus](#) segítségével.
- 2 Az így generált kezdősebességek átlaga nem egzaktul 0, a tömegközéppont sebességével korrigálni kell a kezdőértékeket:
 $\mathbf{v}_i \mapsto \mathbf{v}_i - \mathbf{v}_{\text{CM}}$, ahol a tömegközépponti sebesség:

$$\mathbf{v}_{\text{CM}} = \frac{\sum_{i=1}^N m \mathbf{v}_i}{\sum_{i=1}^N m}.$$

- 3 Ezután az 1 szórású sebességeket a kívánt sebességre kell átskalázni úgy, hogy a kívánt T hőmérsékletet kapjuk: $\mathbf{v}_i \mapsto \lambda \mathbf{v}_i$, ahol

$$\lambda = \sqrt{\frac{2(N-1)k_B T}{\sum_{i=1}^N m v_i^2}}.$$

A program legidőigényesebb – $\mathcal{O}(N^2)$ – része a páronkénti erők kiszámítása. L. Verlet (*Phys. Rev.* **159**, 98 (1967)) cikkében két gyorsítást vezet be:

❶ *A potenciál levágása:*

A Lennard-Jones-erő rövid hatótávú, nagysága $r > r_0$ után gyorsan csökken. Bevezethető r_{cutoff} *levágási távolság*, amin túl a hatása praktikusán 0-nak tekinthető.

Ha r_{cutoff} távolságon belül K részecske van, és $N = nK$, akkor a skálázás teszőleges n -re $\mathcal{O}(nK^2)$ rendű.

Másként fogalmazva, a sűrűséget állandónak tartva n -ben nem négyzetes, hanem lineáris a skálázás.

A módszer hasznos, ha $r_{\text{cutoff}} \ll L$, ahol L a rendszer mérete.

L. Verlet másik gyorsítási javaslata:

② *A szomszédsági lista:*

A levágás érvényesítéséhez tudni kell, melyek azok a szomszédok, amelyekre $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j| < r_{\text{cutoff}}$. Mivel minden részecske mozog, ennek kiértékelése is $\mathcal{O}(N^2)$ feladatnak tűnik.

Ha figyelembe vesszük, hogy egy-egy lépésben a részecskék keveset mozdulnak el, választhatunk egy $L \gg r_{\text{max}} > r_{\text{cutoff}}$ távolságot, amelynél jobban néhány lépésen belül nem távolodnak el a részecskék, és elég azon belül nyilvántartani a szomszédsági listát, és csak néha frissíteni azt.

Verlet javaslata a paraméterválasztásra

- $r_{\text{cutoff}} = 2,5r_0$,
- $r_{\text{max}} = 3,2r_0$, és
- 10 lépésenként frissíteni a szomszédsági listát.

A frissítés gyakorisága és r_{max} közötti összefüggés a sebességek valószínűségeloszlásából becsülhető.

A fentiek megfontolások bevezetésével Verlet 10-szeres sebességnövekedést ért el a pontosság jelentős romlása nélkül.

A levágás kisebb hibákat behoz a szimulációba és elrontja az energia megmaradást, ami korrigálható a potenciál módosításával.

Módosított LJ-potenciál

$$V_{\text{corr}}(r) = V(r) - \left. \frac{dV}{dr} \right|_{r_{\text{cutoff}}} (r - r_{\text{cutoff}}).$$

Ez a kiadott kódokban nincs leprogramozva.

- A hőmérséklet. Egyensúlyban teljesül az ekvipartíció tétele, és a kinetikus energiából származtatható a hőmérséklet. A Lennard-Jones-erő konzervatív, a rendszer teljes energiája megmarad.

$$3(N - 1) \cdot \frac{1}{2} k_B T = \left\langle \frac{m}{2} \sum_{i=1}^N v_i^2 \right\rangle.$$

$\langle \dots \rangle$ jelöli a termikus sokaság átlagot. $3(N - 1)$ a belső szabadsági fokok száma, a tömegközéppont mozgása nem járul hozzá a termikus energiához, ha az átlag sebesség $\bar{\mathbf{v}} \neq 0$, akkor igazából v_i^2 helyett $(\mathbf{v}_i - \bar{\mathbf{v}})^2$ szerepel a kifejezésben.

- Nem-egyensúlyi helyzetben nem igaz az ekvipartíció, de lemérhetjük ezt a mennyiséget, a *pillanatnyi hőmérsékletet*.

- A teljes energia (a mozgási és a potenciális energiák összege):

$$E = \frac{m}{2} \sum_{i=1}^N \mathbf{v}_i^2 + \sum_{i \neq j} V(|\mathbf{r}_i - \mathbf{r}_j|)$$

- A hőkapacitás (a fluktuáció-disszipáció tétel alapján):

$$C_V = \left(\frac{\partial E}{\partial T} \right)_V = \frac{1}{k_B T^2} \left[\langle E^2 \rangle - \langle E \rangle^2 \right]$$

Az átlagolás sokaságátlag – több különböző véletlen futtatás eredménye, de egyensúlyban ezt jól közelíti az időátlag.

A hőkapacitást a hőmérséklet-fluktuációkból is becsülhetjük:

$$\langle T^2 \rangle - \langle T \rangle^2 = \frac{3}{2} N (k_B T)^2 \left[1 - \frac{3 N k_B}{2 C_V} \right].$$

- A nyomást véges zárt rendszernél mérhetnénk a doboz falánál történt impulzusváltozásokból is, de a **Viriál-tétel** alapján a következő kifejezés is használható:

$$PV = Nk_B T + \frac{1}{3} \left\langle \sum_{i < j} \mathbf{r}_{ij} \cdot \mathbf{F}_{ij} \right\rangle.$$

- A **kompesszibilitási faktor** kifejezi, mennyire vagyunk távol az ideális gáz állapottól:

$$Z = \frac{PV}{Nk_B T} = 1 + \frac{1}{3Nk_B T} \left\langle \sum_{i < j} \mathbf{r}_{ij} \cdot \mathbf{F}_{ij} \right\rangle.$$

Nagy sűrűségen a taszító potenciál miatt $Z > 1$, kisebb sűrűségeken pedig a vonzó van der Waals-erők miatt $Z < 1$. Szabad részecskékre, ideális gázra lenne $Z = 1$.

- `md-gl.cpp` 2 dimenziós gáz szimulációja a velocity-Verlet algoritmust használja periodikus határfeltételek mellett. Folyamatosan méri a sebesség-hisztogramot, és összeveti a Maxwell-Boltzmann-eloszlással. Az egérgombok segítségével indítható/állítható a szimuláció, illetve növelhető/csökkenthető a hőmérséklet.
- `md.cpp` program a legegyszerűbb 3 dimenziós Lennard-Jones-szimuláció. Nem használ periodikus határfeltételeket, a velocity-Verlet algoritmussal léptet, és a pillanatnyi hőmérsékletet egy fájlba menti ki.
- `md2.cpp` program tartalmazza a sebességek inicializálását, figyeli a pillanatnyi hőmérsékletet, és amennyiben nem felel meg a megjelölt értékek, újra átskálázza a sebességeket.
- `md3.cpp` program 3 dimenziós rendszer szimulátora, tartalmazza a Verlet-féle közelítéskereszt és gyorsítást.
- `md3-gl.cpp` program az előző OpenGL ábrázolással kiegészített változata.

- 1 Értsük meg az `md.cpp` programot, ábrázoljuk kimenetét, és értelmezzük a kapott görbét! Vizsgáljuk meg az `md2.cpp` programban a javításokat. A program futtatási eredményét elemezve vizsgáljuk meg, hogy mennyit segített a kezdeti feltételek pontosabb beállítása, és mennyi idő alatt áll be a kívánt hőmérséklet!
- 2 Értsük meg a `md3.cpp` program hogyan kezeli a levágásokat, a szomszédsági listát, és annak frissítését. Kísérletezzünk más r_{cutoff} , r_{max} és `updateInterval` paraméterekkel. Nézzük meg hogyan változik a sebesség és a pontosság az eredeti `md2.cpp` programhoz képest.
- 3 Írjuk meg a teljes energiát, nyomást, hőkapacitást, kompresszibilitási faktort számoló függvényeket! Próbáljuk meg a fázisátalakulási hőmérsékletet meghatározni a rendparaméterek (hőkapacitás, kompresszibilitás) figyelésével. Vessük össze az értékeket a valódi **argon** olvadás és forráspontjával! Hogyan befolyásolja N értéke a tapasztaltakat?

Lapozz!

- 4 Alakítsuk át a programot úgy, hogy a periodikus határfeltétel helyett zárt, kemény fala legyen, ahonnan visszapattannak a részecskék! (Egyszerű megoldás a sebességek és pozíciók tükrözése, kicsit bonyolultabb a falat is tömör Lennard-Jones potenciállal leírható anyagnak venni.) Mérjük a nyomást a falakra kifejtett erő segítségével, és ellenőrizzük mennyire teljesül a $pV = Nk_B T$ törvény!
- 5 Értsük meg a md3-gl.cpp programban a 3D grafika használatát! Szerkesszük össze a bolygómozgás szimulátorával, hogy a gravitációs háromtestes- (vagy soktestes-) probléma dinamikáját ábrázolja. Vajon alkalmazható itt is egy levágási hossz a kód gyorsítására?