

Molekuladinamika

Tóth Balázs



Természettudományi kar
Eötvös Loránd Tudományegyetem
Magyarország
2020 április 24

1 Elméleti bevezetés

Molekuladinamika alatt nagy számú részecskék dinamikai vizsgálatát érjük. A mai technológia azonban nem engedi a potenciálisan 10^{23} nagyságrendű részecskeszám mozgásának lekövetését, ezért közelítéseket kell alkalmaznunk.:

- A távoli erők nagyobb térrészekre kiátlagolódnak
- A nagyon távoli részecskék elhanyagolhatók

A rendszer szimulációja során N darab atomunk van, amiknek mindegyikének ismert a kezdő sebessége és pozíciója. Miközben a részecskéket időben léptetjük a Newton-törvényeket felhasználva páronként kiszámoljuk az erőt. Ezek közben figyeljük, hogy a rendszer a kezdeti tranziens után termodinamikai egyensúlyba kerül-e. Az egyensúly fennállása esetén használhatjuk az ekvipartíció tételét:

$$\text{tartalom...} \langle K \rangle = \langle \frac{1}{2} m V^2 \rangle = \frac{3}{2} k_b T$$

Felmerülhet bennünk a kérdés, hogy a szimuláció során nem-e kell figyelembe vennünk a kvantummechanikát, de beláthatjuk, hogy ha nemesgázokat modellezünk, akkor a klasszikus közelítés jó képet fest:

- A $T=300\text{K}$ hőmérsékleten az Ar gáz részecskéinek mozgási energiája sokkal kisebb mint az elektronok gerjesztési energiája.
- A vizsgált atomok de Broglie-hullámhossza

Érdeemes még kitérni arra a problémára, hogy érdemes a szimuláció kezdeti feltételeit a termodinamikai egyensúly közelébe állítsuk be, ellenkező esetben a nagy számítási költségek miatt gondot okozhat az egyensúly elérése.

2 Szimuláció leírása

A molekuladinamika vizsgálatához a szimulációnk során két elterjedt algoritmust használtam fel:

- Verlet-algoritmus
- velocity-Verlet-algoritmus

A következő részben röviden kívánom ismertetni a módszereink előnyeit valamint a hátrányait:

2.1 Verlet-algoritmus

Az algoritmus léptető számályai:

$$\vec{R}_{n+1} = 2\vec{R}_n - \vec{R}_{n-1} + \tau^2 \vec{A}_n + \mathcal{O}(\tau^4)$$

$$\vec{V}_n = \frac{\vec{R}_{n+1} - \vec{R}_{n-1}}{2\tau} + \mathcal{O}(\tau^2)$$

Előnyei:

- Gyorsabb mint a Runge-Kutta-módszer
- Majdnem olyan pontos, mint a negyedrendű Runge-Kutta-módszer
- Jól megőrzi az energiát
- Ergodikus: időtükrözésre nem változik

Hátrányai

- Három pontos rekurzió
- A sebesség hibája
- A sebesség és a pozíció nem egyszerre frissül, a sebesség lemarad

2.2 velocity-Verlet-algoritmus

Az algoritmus léptető számályai:

$$\vec{R}_{n+1} = \vec{R}_n - \tau \vec{V}_n + \frac{\tau^2}{2} \vec{A}_n + \mathcal{O}(\tau^3)$$

$$\vec{V}_{n+1} = \vec{V}_n + \frac{\tau}{2} (\vec{A}_{n+1} + \vec{A}_n) + \mathcal{O}(\tau^3)$$

Előnyei:

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

Hátrányai

- R-ben pontatlanabb

A modellünkben a Lennard-Jones-potenciál negatív gradienséből kapjuk meg az erős és a teljes szimulált rendszer energiáját:

$$E = \sum \frac{1}{2} m v_i^2 + \sum V(r_{ij})$$

Egyensúly esetén a sebességeknek a Maxwell-Boltzmann-eloszlást kell követniük. Ha N darab Ar atomot szeretnénk vizsgálni egy négyzetbe (2D), akkor megadhatjuk a formulában szereplő V_0, K, r_0 értékeket, valamint a rendszer karakterisztikus idejét.

Ezen kívül érdemes még átváltanunk SI mértékegységekről, mivel az értékeink túlságosan picik. At új rendszerünk egy olyan skalárrendszer, ahol $m = r_0 = V_0 = \tau_0 = 1$.

A program legidőigényesebb része nyilván az atompárok között fellépő erők kiszámítása. Ennek gyorsítására két jól bevált módszert használunk:

- A potenciál levágása: Mivel az erő rövid hatótávolságú ezért bevezethető egy levágási távolság, ami felett az erő értékeit nullának vesszük
- Szomszédsági lista: Ha azt vesszük, hogy az atomok lépésenként csak kis mértékben mozdulnak el, akkor nyilvántarthatjuk a szomszédos atomokat és csak néha frissítjük ezt a listát.

3 Termodinamikai mennyiségek

3.1 Hőmérséklet

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

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

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

3.2 Teljes energia

A teljes energia a mozgási és a potenciális energiák összege.

$$E = \frac{m}{2} \sum_{i=1}^N v_i^2 + \sum_{i \neq j} V(|r_i - r_j|)$$

3.3 Hőkapacitás

A fluktuáció-disszipáció tétele alapján a következő képen néz ki:

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

3.4 Nyomás

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

$$PV = Nk_B T + \frac{1}{3} \langle \sum_{i < j} r_{ij} F_{ij} \rangle$$

3.5 A kompresszibilitási faktor

A kompresszibilitási faktor kifejezi, mennyire vagyunk távol az ideális gáz állapottól:

$$Z = \frac{PV}{Nk_B T} = 1 + \frac{1}{3Nk_B T} \langle \sum_{i < j} r_{ij} F_{ij} \rangle$$

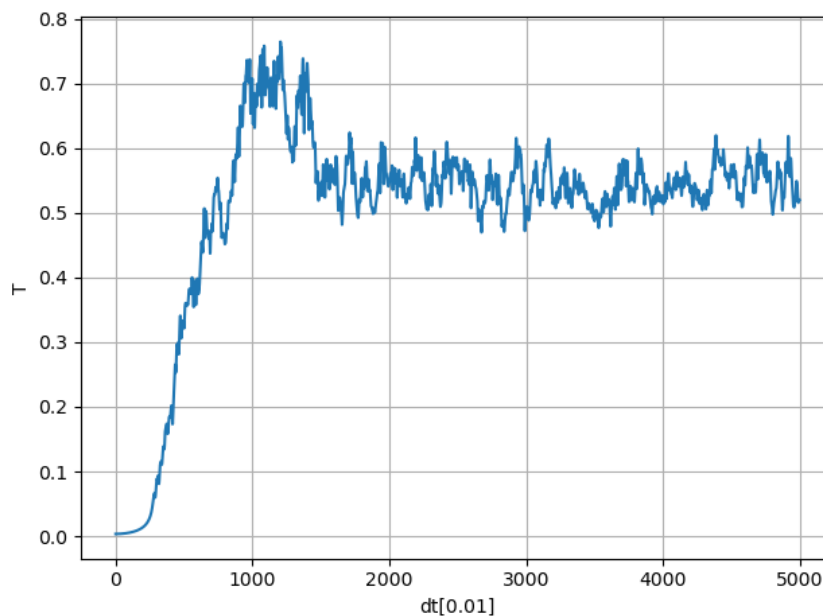


Figure 1: Az md.cpp által kiszámolt pillanatnyi hőmérsékletek

4 Eredmények ismertetése

4.1 1.feladat

Az első feladatunk az volt, hogy megértsük és összehasonlítsuk az md1.cpp és az md2.cpp működését. A program a velocity-Verlet-algoritmussal számol minden egyes részecskepárra erőt és hőmérsékletet minden egyes lépés után. Ahhoz, hogy jól értelmezhetőek legyenek az eredmények a szimulációt 5000 léptetésig futtattam és ábrázoltam. Megfigyelhetjük, hogy nagyjából a 2000. lépés körül beáll az egyensúly és már csak fluktuálunk körülötte. A második scriptben több javítás is felfedezhető, amik rendre a következők:

- Van periodikus határfeltétel
- Kezdősebesség beállítása egy 0 várható értékű és 1 szórási függvényvel
- Sebességkorrekciót és sebesség átskalázást is tartalmaz
- A kód elején megadott $T=1$ hez konvergálunk az egyensúlyban

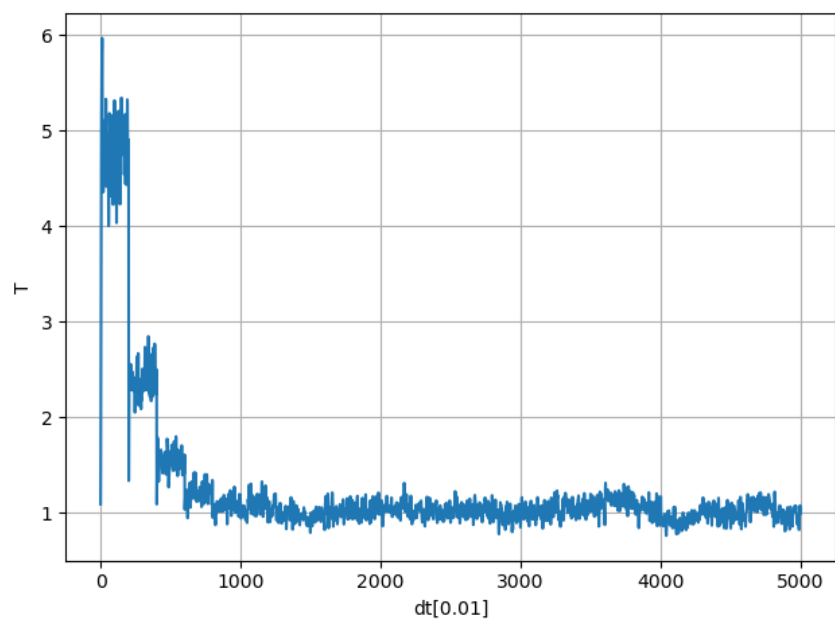


Figure 2: Az md2.cpp által kiszámolt pillanatnyi hőmérsékletek

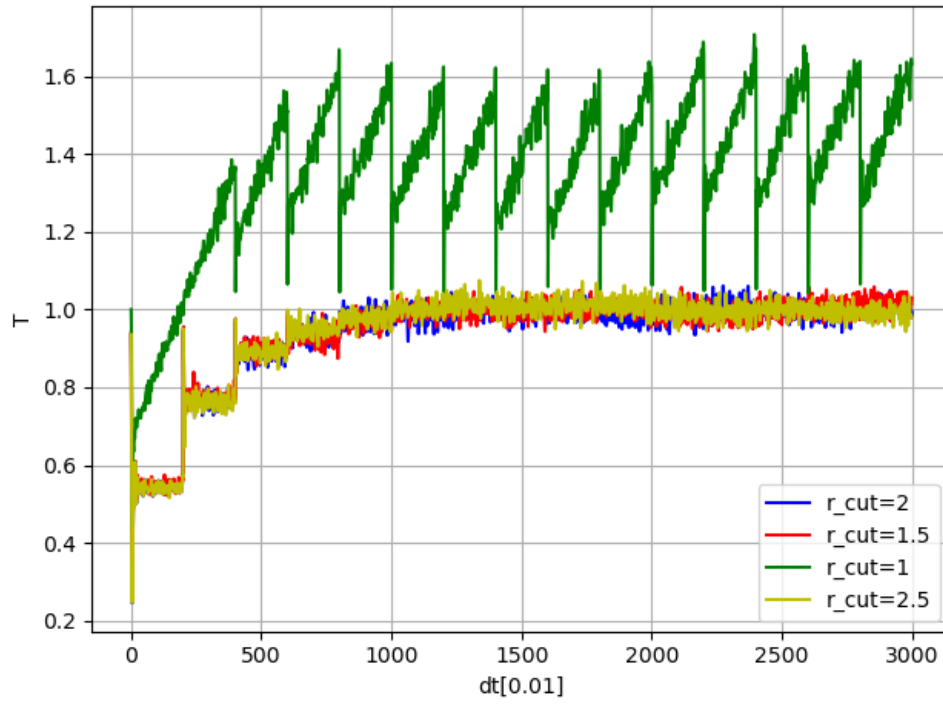


Figure 3: Az md3.cpp által kiszámolt pillanatnyi hőmérsékletek különböző levágási szélességekkel

4.2 2.feladat

A következő feladatban a md3.cpp működését tanulmányoztuk és vizsgáltuk a levágási távolság, a maximális távolság és a frissítési intervallum változtatásának a rendszerre gyakorolt hatását. A tapasztalataink azt mutatják, hogy ha a levágási távolság csökkentésével egyre instabilabbá válik a rendszer és 1 alatt teljesen szét is esik. A maximális távolság csökkentésével pedig csökken a hőmérséklet valamint növekszik a fluktuáció. Az updateinterval változtatása szemmel láthatóan nincs nagy hatással a kód működésére.

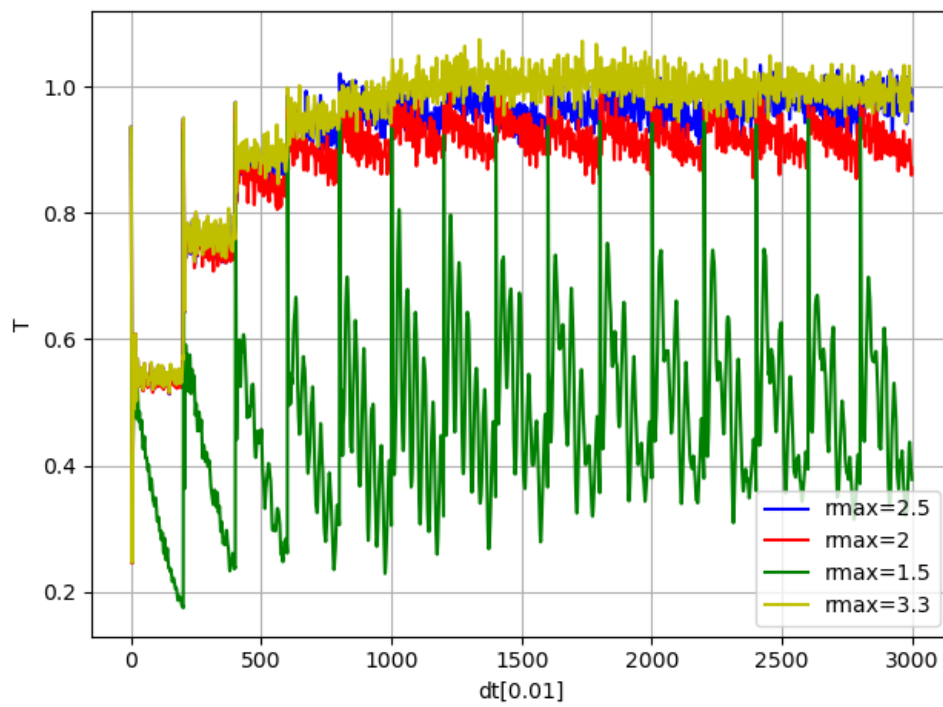


Figure 4: Az md3.cpp által kiszámolt pillanatnyi hőmérsékletek különböző maximális szélességekkel

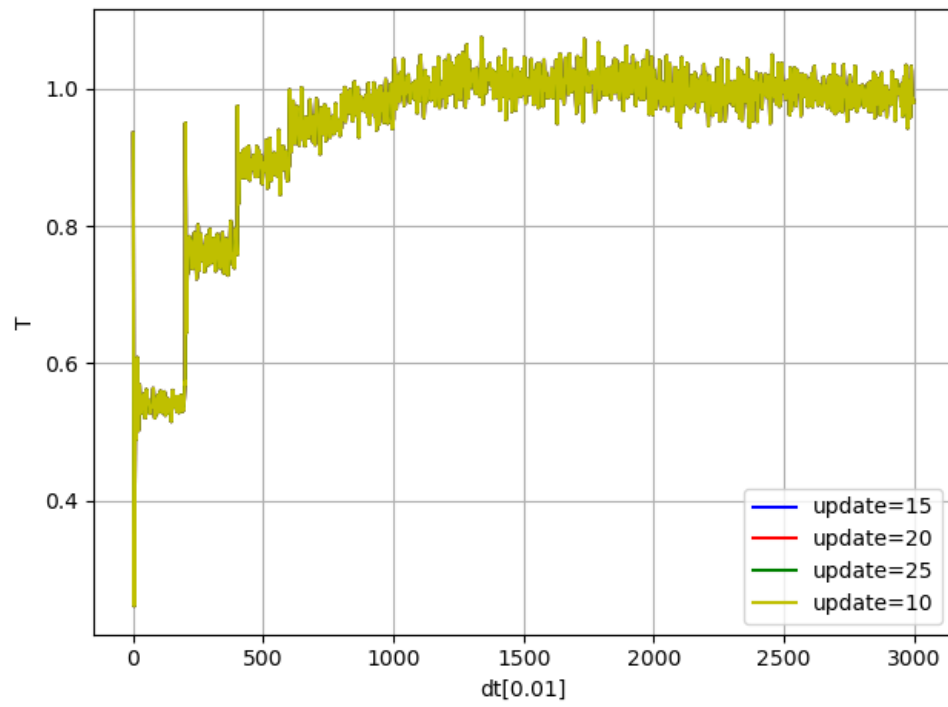


Figure 5: Az md3.cpp által kiszámolt pillanatnyi hőmérsékletek különböző frissítési intervallummal

4.3 3.feladat

Ennél a feladatnál az volt a feladatunk, hogy a megadott kódot úgy egészítsük ki, hogy a következő értékeket is kiszámolja nekünk:

- Teljes energia
- Nyomás
- Hőkapacitás
- Kompresszibilitási faktor

A vizsgált mennyiségeket egyenként ábrázoltam is (6,7,8,9 ábrák). Azt tapasztaltam, hogy az energia, valamint a térfogat és nyomás szorzata hasonló görbét rajzolnak ki mint a korábbi szimulációk.

A hőkapacitáson azt láthatjuk, hogy az elején nagyon megugrik, azonban gyorsan stabilizálódik.

Végül a kompresszibilitási faktor végig konstans érték marad. A feladat végén megvizsgáltam, hogyan változik a kimenet, ha a részecskeszámot több mint kétszeresére emelem ($N=2000$). Azt tapasztalata, hogy a kompresszibilitás és a nyomás nem változott, azonban a hőkapacitás a felére csökkent az energia pedig több mint a kétszeresére növekedett.

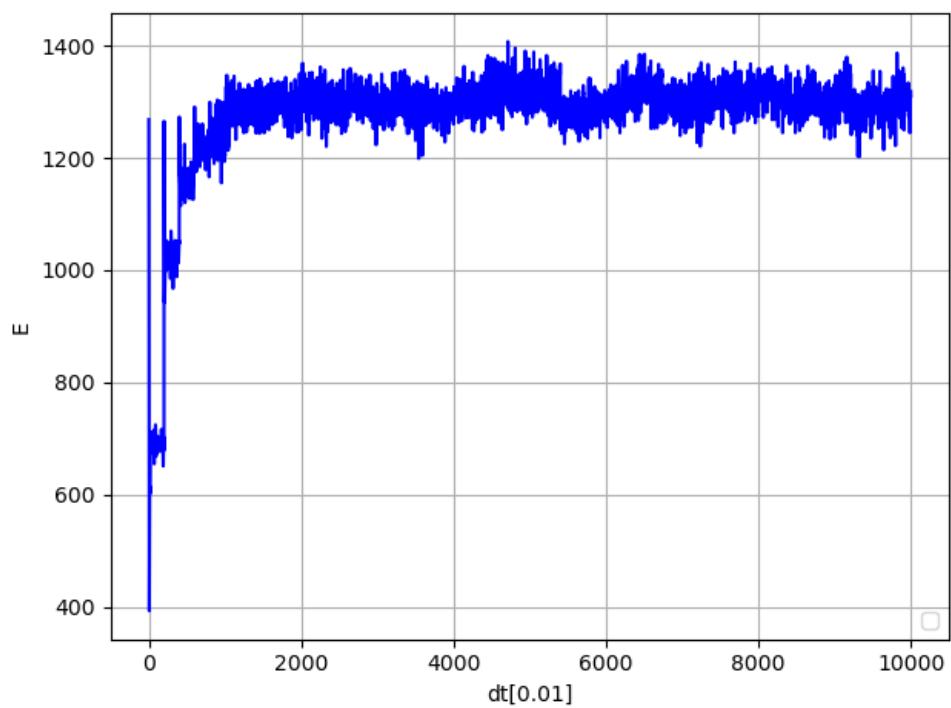


Figure 6: md3.cpp által kiszámolt pillanatnyi energia az egyes léptetésekre

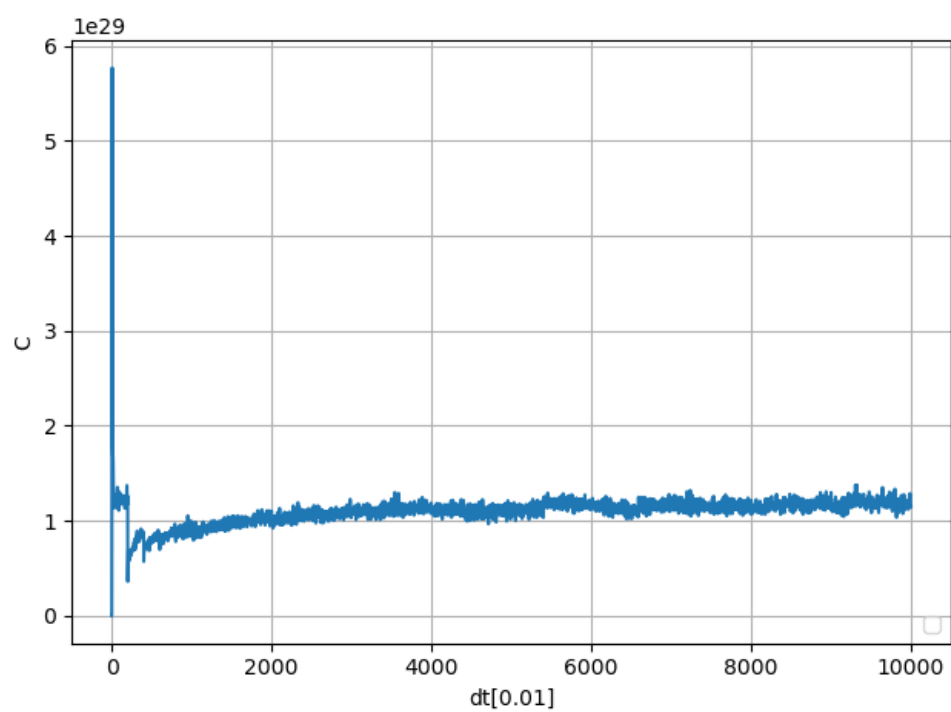


Figure 7: md3.cpp által kiszámolt pillanatnyi hőkapacitás az egyes léptetésekre

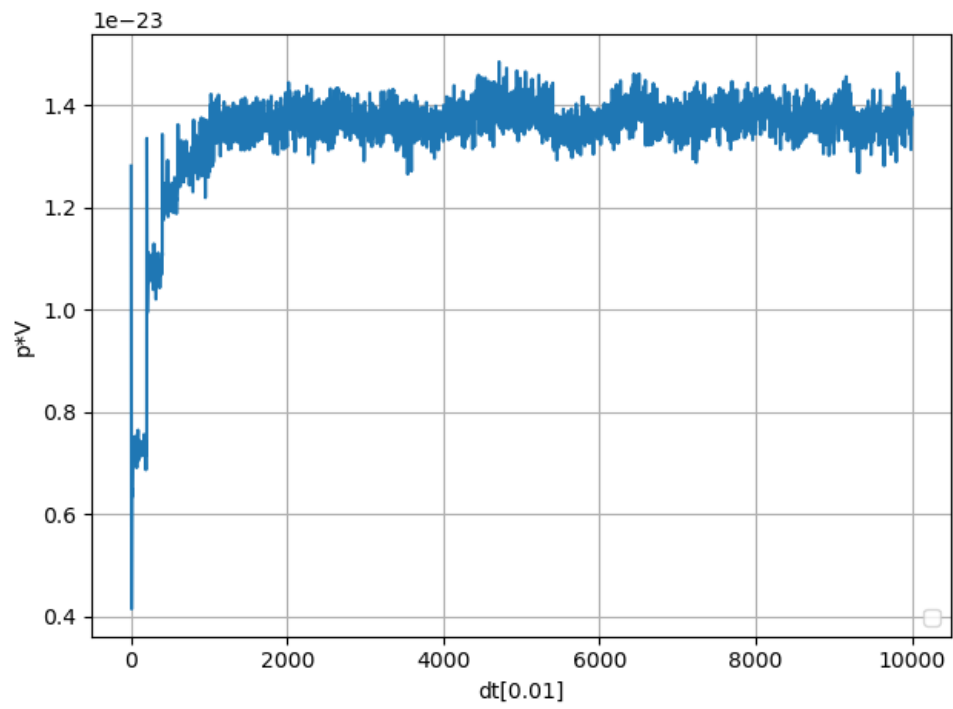


Figure 8: md3.cpp által kiszámolt pillanatnyi nyomás megszorozva a térfogattal az egyes léptetésekre

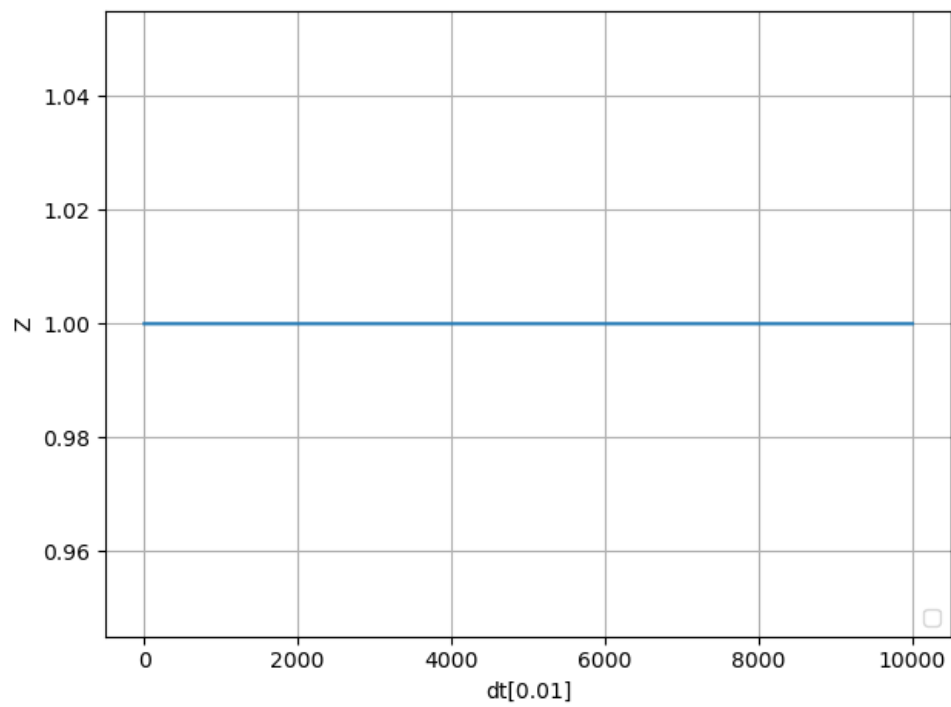


Figure 9: md3.cpp által kiszámolt pillanatnyi kompresszibilitás az egyes léptetésekre

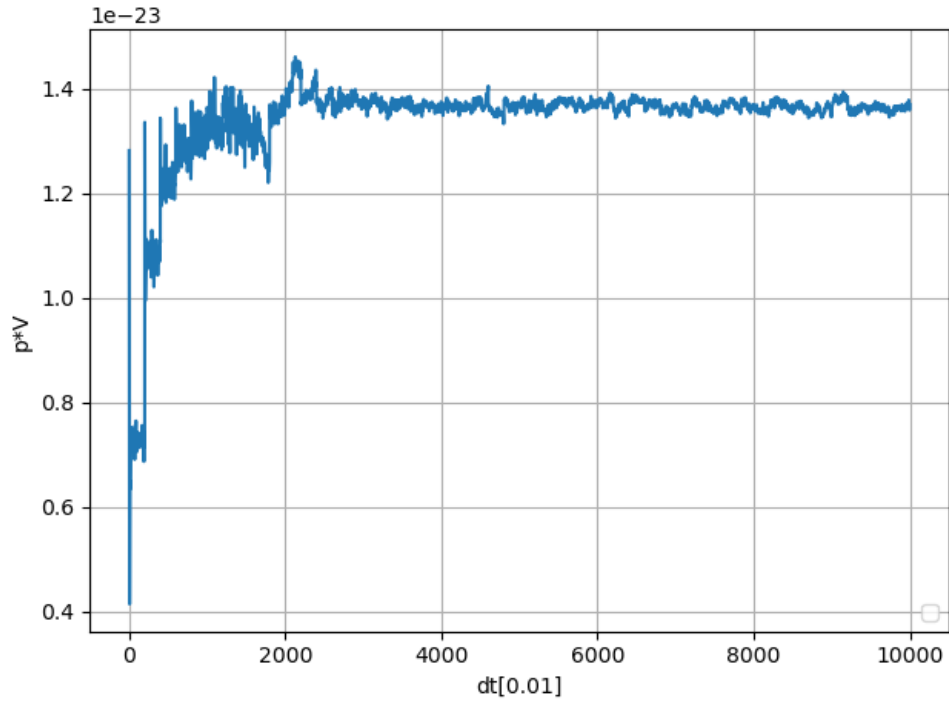


Figure 10: md3.cpp által kiszámolt pillanatnyi nyomás megszorozva a térfogattal az egyes léptetésekre kemény fal esetén

4.4 4.feladat

Ez a feladat arról szólt, hogy át kellett alakítanunk a kódot úgy, hogy a periodikus határfeltétel helyett zárt, kemény fala legyen amiről a részecskék visszapattannak. Ezt legegyszerűbben úgy valósíthatjuk meg, ha a határfeltételeknél tükrözzük a részecskék sebességet és a pozíciókat.

A feladat során a nyomás és a térfogat szorzatát leosztottam a részecskeszámmal és a Boltzmann állandóval, így vizsgálni tudtam a $pV = Nk_bT$ összefüggés teljesülését. Az eredmény megegyezett a korábbi szimulációk hőmérsékletével azonban el volt csúsztatva három nagyságrenddel.

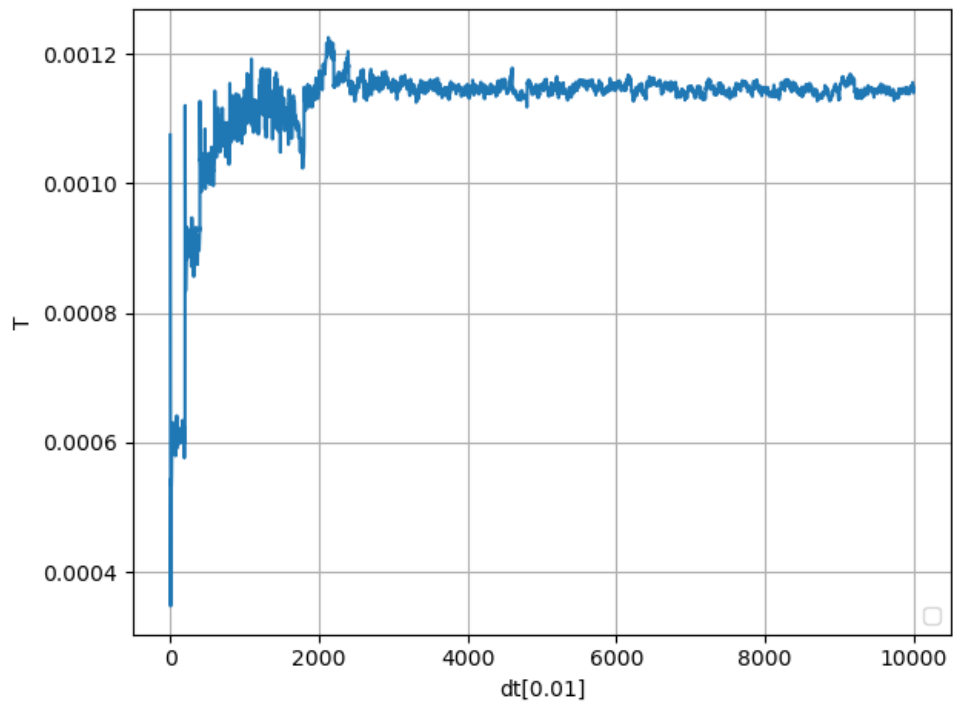


Figure 11: md3.cpp által kiszámolt nyomás és térfogat szorzatából származtatott hőmérséklet egyes léptetésekre kemény fal esetén

5 Konklúzió

A jegyzőkönyvben bemutattam a molekuladinamika szimulációs problémájának egy egyszerű változatát és annak megvalósítását. Ezen kívül még ismertettem az egyensúlyi hőmérséklet stabilizálódásához szükséges idő függését. Tárgyaltam az egyensúlyi hőmérséklet fluktuációjának a függését a levágási hossz, maximális hossz valamint az `updateInterval` paraméterektől a Verlet algoritmusban. A szimuláció kiértékelése során vizsgáltam még a különböző termodinamikai mennyiségek időfüggését.