Molekula dinamika

Szőke Kálmán Benjamin - SZKRADT.ELTE

2012. április 26.

1. Bevezetés

Ebben a jegyzőkönyvben a molekula dinamika modellezése volt a feladat. A forráskód a szimulációhoz elérhető volt a kurzus honlapján.

2. Elméleti leírás

Molekula dinamika során a részecskék mikroszkopikus dinamikáját követjük nyomon, ezek jellemző darabszáma $6 \cdot 10^{23}$ nagyságrendben van. Számítógéppel ekkora nagyságrendet nem tudunk szimulálni, azonban elég sok részecskét követhetünk, amelyekre már értelmezhetők a termodinamika törvényszerűségei. Közelítéseket alkalmazva egyre több részecske modellezése oldható meg, azonban végig vigyáznunk kell, hogy a fizikai valóságtól ne rugaszkodjunk el, csak azokat a feltételeket hanyagoljuk el, amelyek a rendszerben amúgy is elhanyagolható nagyságúak. Legyen tehát N számú atomunk, melyeknek a szimuláció kezdetén ismerjük kezdő helyzetüket és sebességüket. A Newton-törvények értelmében a részecskék közt páronként számoljuk ki az erőt, majd léptetjük a részecskét. A rendszer egyensúlyának vizsgálata lényeges, ilyenkor teljesül az ekvipartíció tétele.

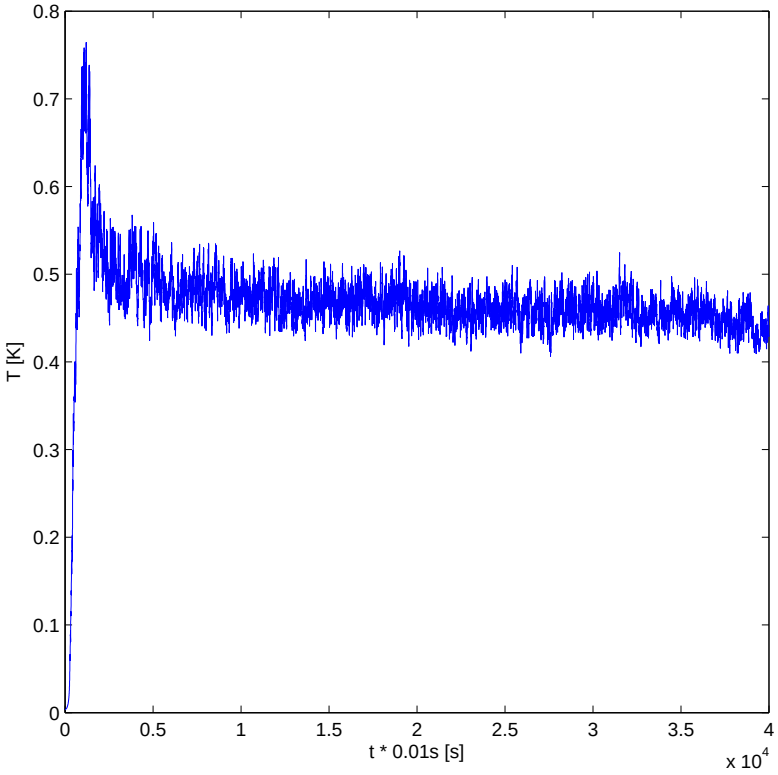
$$\langle K \rangle = \left\langle \frac{1}{2} m v^2 \right\rangle = \left\langle \frac{3}{2} kT \right\rangle$$

Itt k a Boltzmann állandó vizsgálatával pedig arról kaphatunk információt, hogy a rendszer egyensúlyban van-e. Ez utóbbi pontos ismerete tehát lényeges, hiszen ilyenkor mérhető a hőmérséklet (T), a nyomás (p) és a többi termodinamikai állapotjelző.

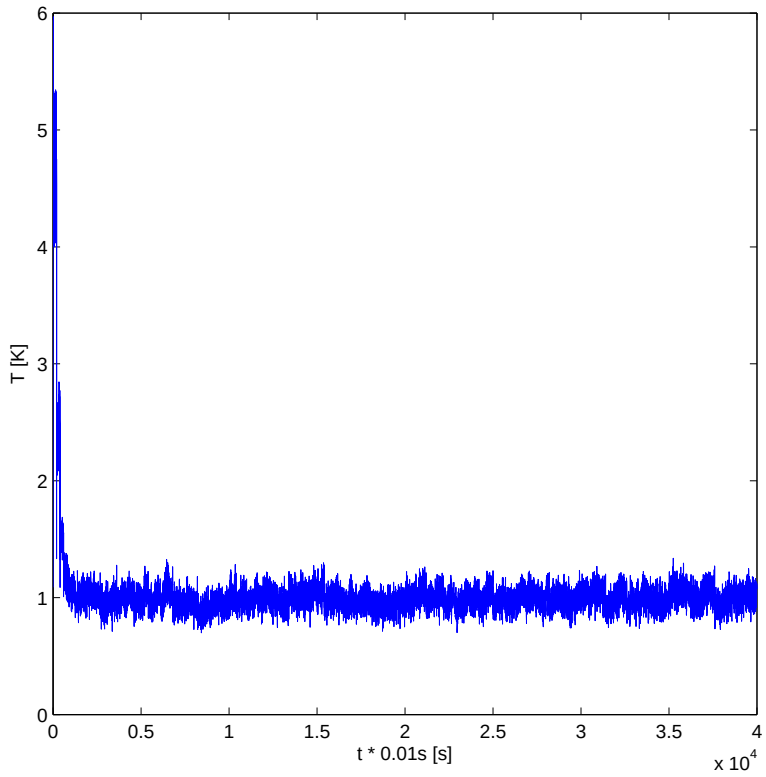
3. Eredmények

3.1. 1. Feladat (md.cpp, md2.cpp)

Az md.cpp és az md2.cpp-vel kapott eredményeket kellett összehasonlítani a feladat során. Az md2.cpp a kezdeti sebességeket Maxwell-Boltzmann eloszlás szerint generálja, figyeli a pillanatnyi hőmérsékletet, és amennyiben nem felel meg a megjelölt értéknek, újra átskálázza a sebességeket. Ezeket figyelembe véve látszik az ábrákon hogy az md2.cpp-vel kapott eredményben az egyensúlyba való beállás relaxációs ideje csökken, de a kezdeti fluktuációk nagyobbak az algorítmus javítása miatt.



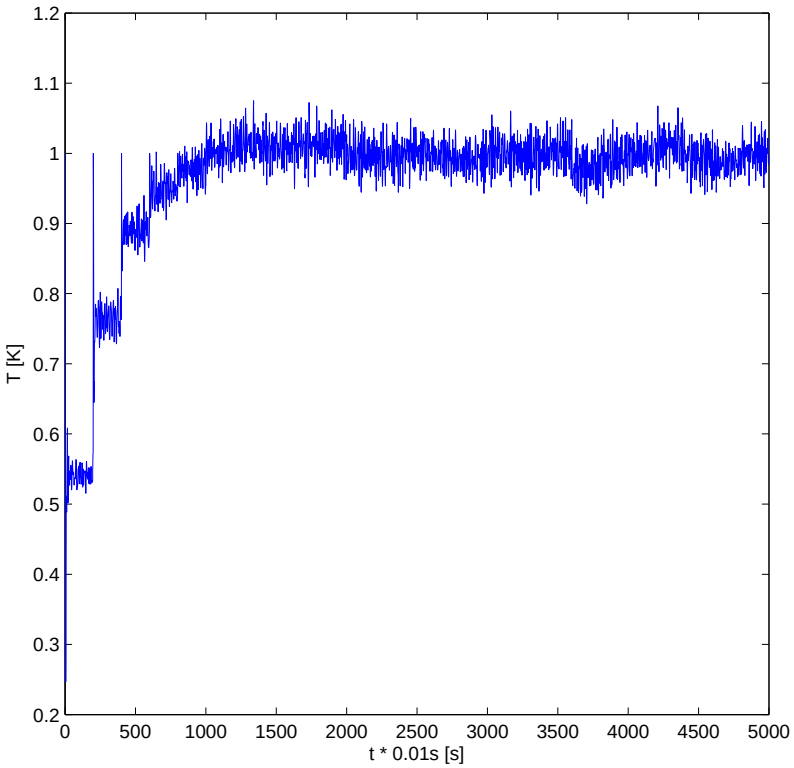
1. ábra. Hőmérséklet md.cpp



2. ábra. Hőmérséklet md2.cpp

3.2. 2. Feladat (md3.cpp)

Az md3.cpp javításokat is alkalmaz a potenciál egyszerűsítésével. Egy adott r_{cutoff} levágási távolság után 0-vá teszi a potenciált, így gyorsabb az algoritmus. Ehhez tudni kell, hogy egy részecske mikor kerül az r_{cutoff} távolságon belülre. Emellett elég egy másik r_{max} távolságon belül figyelni a szomszédos atomokat, ezaz r_{max} a néhány lépés utáni max megtett úthossz. Az r_{cutoff} értéket módosítva észrevejető, hogy ha csökkentjük az értékét akkor a futási idő is csökken. A clock() függvénnyel mérve az időt $r_{cutoff} = 2.5$ értéknél a futási idő: $187.683\,s$, $r_{cutoff} = 2$ esetén $79.653\,s$.

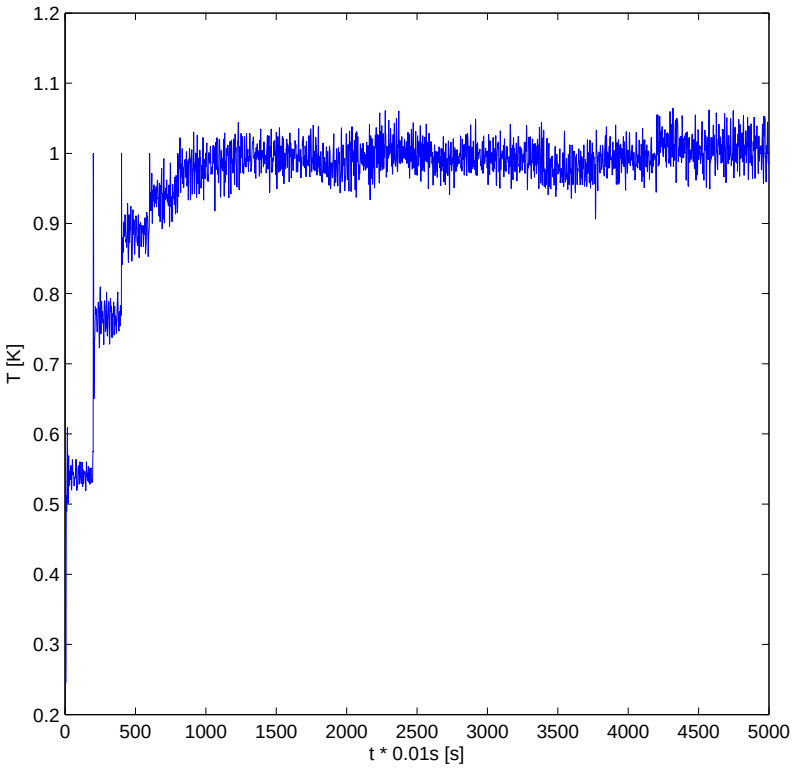


3. ábra. Hőmérséklet md3.cpp

3.3. 3. Feladat (energia, nyomás, hőkapacitás, kompresszibilitás)

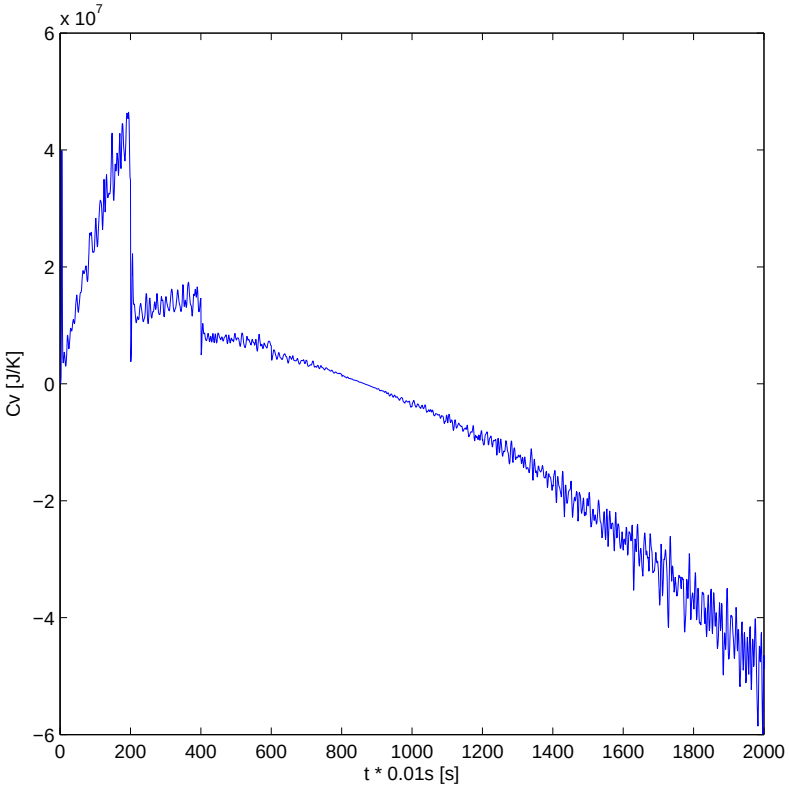
A feladatok számolásához módosításokra volt szükség. Ezek a fájl kiíratás résznél megtalálhatók az alábbi kódrészletben.

```
int main() {
 clock_t start = clock();
 initialize();
 updatePairList();
 updatePairSeparations();
 computeAccelerations();
 double dt = 0.01;
 ofstream file("T3.dat");
 file << "%% Energy" << "\t" << "T" << "\t" << "P" << "\t" << "Z" << "\t" << "Cv" << endl;
 double a_sum = 0;
 double out_T = 0;
 double E = 0;
 double E_2 = 0;
 double Energy = 0;
```

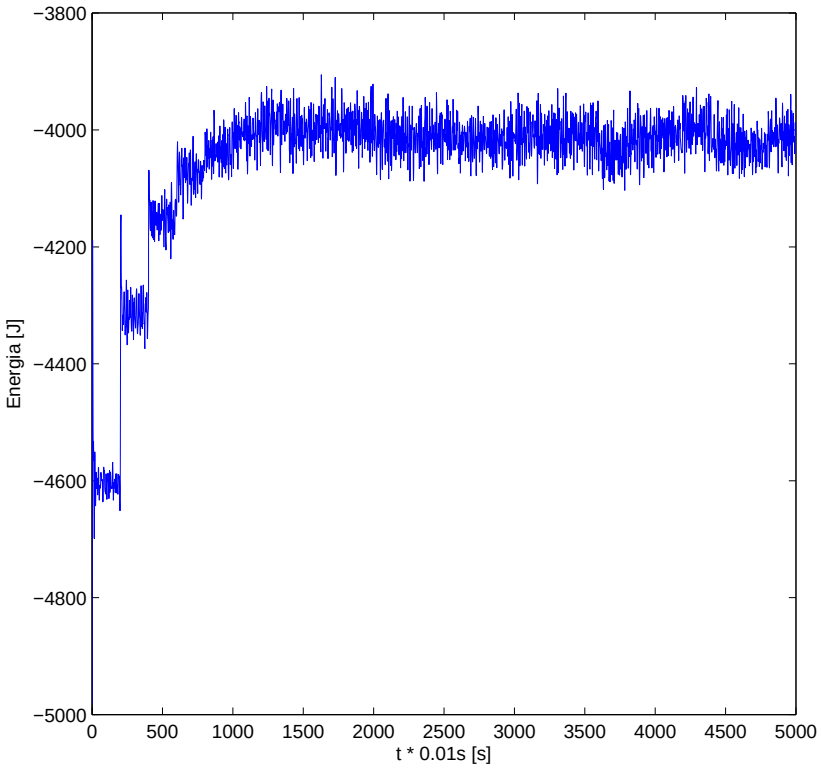


4. ábra. Hőmérséklet md2.cpp $r_{cutoff} = 2$, $r_{max} = 3$

```
 for (int i = 0; i < 5000; i++) {
  file << (Energy=velocityVerlet(dt, a_sum)) << "\t" << (out_T=instantaneousTemperature()) << "\t" <<
  (N*out_T+(1.0/3)*a_sum)/(pow(L,3)) << "\t" << (N*out_T+(1.0/3)*a_sum)/(N*out_T) << "\t";
  E_2 = pow(Energy, 2);
  E = Energy;
  file << (((E_2/N)-pow((E/N), 2))/(pow(out_T, 2))) << endl;
  if (i % 200 == 0)
  rescaleVelocities();
  if (i % updateInterval == 0) {
  updatePairList();
  updatePairSeparations();
  }
 }
 file.close();
 clock_t stop = clock();
 double time = (double)(stop-start)/CLOCKS_PER_SEC;
 ofstream proc("time.dat");
 proc << "szamitasi ido: " << time;
 proc.close();
}
```

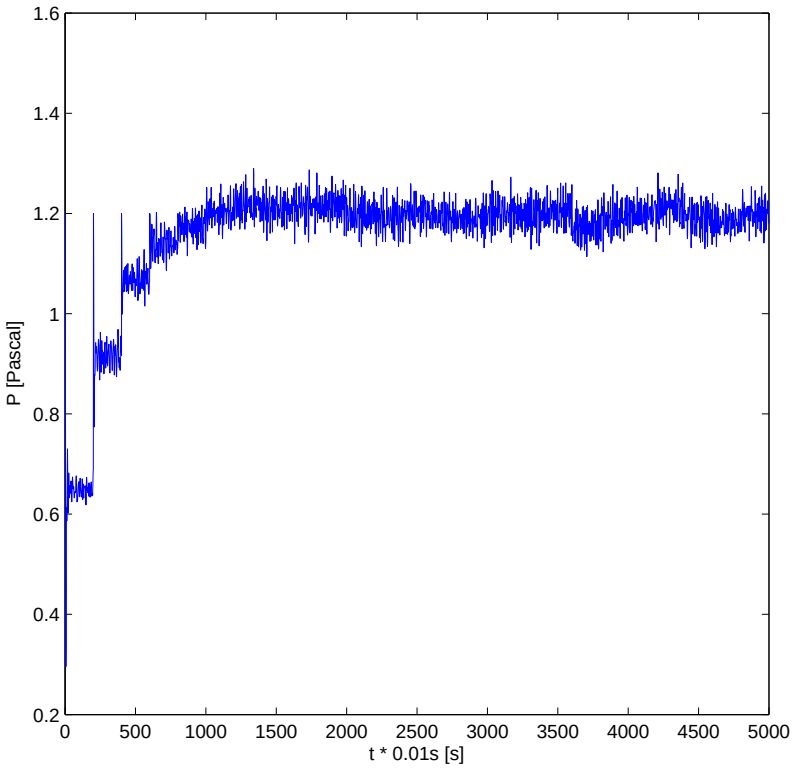


5. ábra. Hőkapacítás md3.cpp



6. ábra. Energia md3.cpp

Tartalomjegyzék

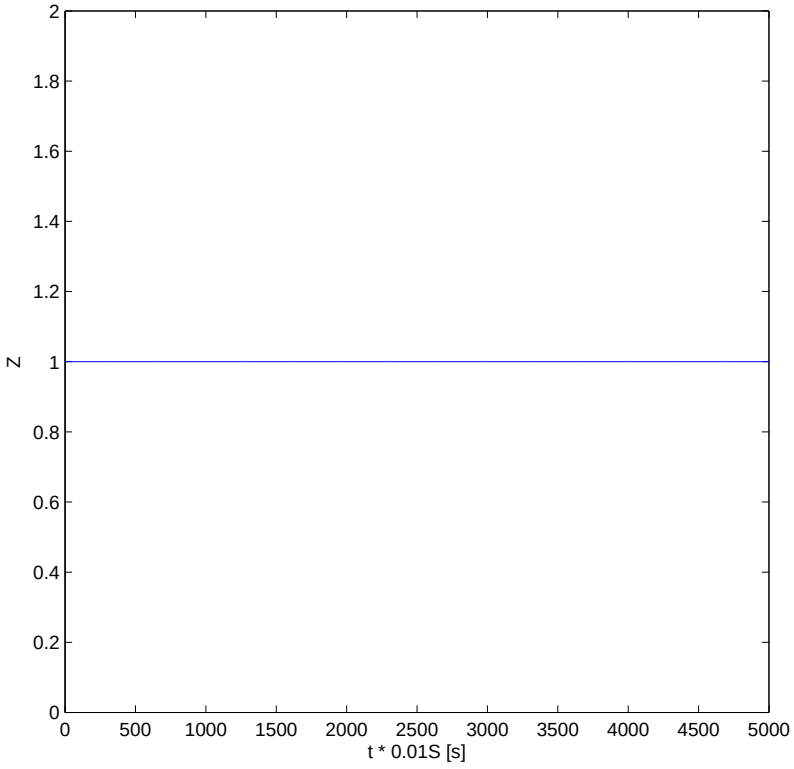


7. ábra. Nyomás md3.cpp

Hivatkozások

[1] Jegyzet
http://complex.elte.hu/∼csabai/szamszim/simLec5.pdf

[2] C++ forráskód
http://complex.elte.hu/∼csabai/szamszim/code/sz2009/md/



8. ábra. Kompresszibilitási faktor md3.cpp